

PTE ÁJK-KTK Könyvtár

KH 1429

# Doktori Értekezés

2003

KRUZSLICZ FERENC



07

650.1

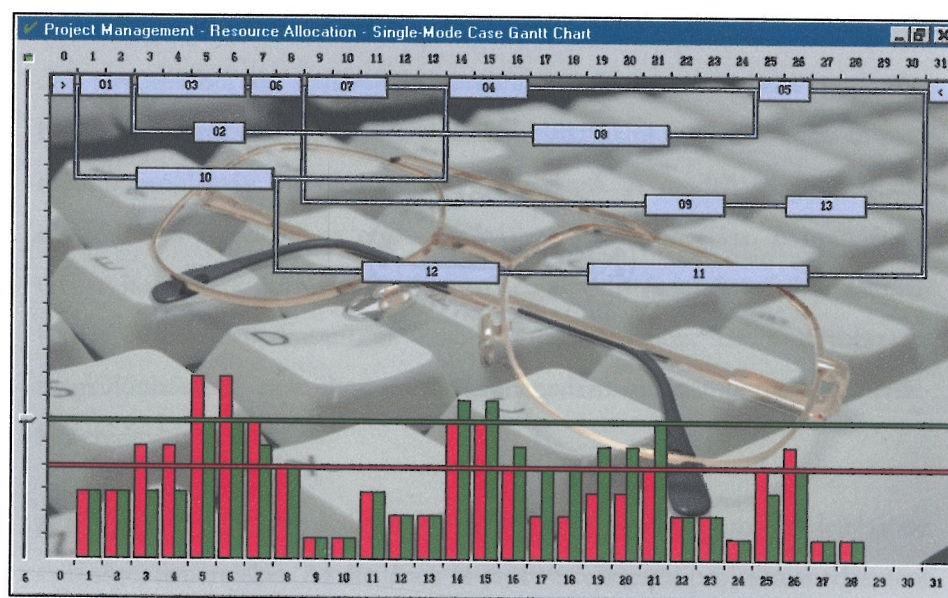
K 96

# SZOFTVERFEJLESZTÉSI PROJEKTEK ÜTEMEZÉSÉNEK MODELLEZÉSE

Kruzslicz Ferenc

PhD disszertáció

Témavezető: Dr. Csébfalvi György



Az értekezés a Közgazdaságtudományi Kar  
Gazdálkodástani Doktori Iskolájában készült.

Pécsi Tudományegyetem

2003

2 melléklettel

PTE Egyetemi Könyvtár



P000818783







Pécsi Tudományegyetem

Kivonat

SZOFTVERFEJLESZTÉSI  
PROJEKTEK  
ÜTEMEZÉSÉNEK  
MODELLEZÉSE

Kruzslicz Ferenc

A dolgozat a projektmenedzsment témakörének egy speciális területét, a szoftverfejlesztési projektek vizsgálatát tűzte ki célul. A középpontban egy olyan újfajta projekt-ütemezési modell áll, amely néhány ismert elméleti modell ötvözéséből és továbbfejlesztéséből keletkezett. A modell gyakorlati haszna is említésre méltó, hiszen a kisebb feladatoktól kezdve az egészen nagy méretű problémák megoldására is alkalmazható. A modell megalkotásának egy lényeges alapeleme az erőforrások újszerű kategorizálása. A gyengén és erősen korlátos erőforrások fogalma erősen kötődik (a szoftverfejlesztési projektek esetében tipikusan) az emberi erőforrásokhoz. Ez a természetes megkülönböztetés az emberi erőforrás menedzsment szempontjából is érdekes eredmény.

A dolgozat a fokozatos szűkítés logikája mentén tárgyalja a modell leírásához szükséges ismereteket. Az általános és szoftverfejlesztési projektmenedzsment fogalmak után a gyenge és erős erőforráskorlátok definiálására kerül sor, először a vezetés, a szervezet és a szervezés területén, majd pedig a matematikai modell szintjén. A modell és a megoldására kidolgozott eljárások ismertetése néhány alkalmazási esettanulmánnyal zárul.



## TARTALOMJEGYZÉK

1. A PROJEKTMENEDZSMENT ALAPFOGALMAI	3
1.1 A projekt	4
1.2 A projektmenedzsment	6
1.3 A projekt életciklusa	9
1.4 Projektek sikertényezői	11
2. EMBERI ERŐFORRÁS ÉS PROJEKTMENEDZSMENT	13
2.1 Team- és projektszervezetek	15
2.2 Az emberi erőforrás menedzsment és az ütemezés kapcsolatrendszere	19
2.3 Erőforrás-tervezés	20
2.4 Ütemezéshez használt becslési módszerek	25
2.5 Erősen és gyengén korlátos (emberi) erőforrások	27
3. INFORMATIKAI PROJEKTEK MENEDZSELÉSE	31
3.1 Szoftver- és rendszerfejlesztési projektek	32
3.2 A szoftverfejlesztési projektek speciális aspektusai	34
3.3 Egyszerűsítések kisebb szoftverfejlesztési projektek esetén	36
4. TEVÉKENYSÉGI HÁLÓK, HÁLÓTERVEZÉS	38
4.1 Időmenedzsment és ütemezés-tervezés	38
4.2 Erőforrás hozzárendelési és kiegyenlítési problémák	41
4.3 Jelölések	44
4.4 A kritikus út módszere	47
4.5 Határidők és tartalékidők	49
5. ERŐFORRÁS KIEGYENLÍTÉSI MILP MODELLEK	50
5.1 Globális erőforrás kiegyenlítő mérték	55
5.2 Vegyes egészértékű programozási alap modell	59
5.3 Erősen és gyengén korlátos erőforrások MILP modellje	68



6. ERŐFORRÁS KIEGYENLÍTÉS IMPLICIT LESZÁMLÁLÁSSAL	77
6.1 Az implicit leszámhlási alapmodell	77
6.2 Leszámlhlási algoritmusok és keresőfák	78
6.3 Alsó becslések és vágási szabályok	81
6.3.1 A „legalább olyan jó” vágási szabály	83
6.3.2 Az „LP relaxálás” vágási szabály	88
6.4 Implicit leszámhlási modell gyenge és erős erőforrások esetén	90
6.5 Kétfázisú implicit leszámhlási algoritmus	91
6.5.1 Első fázis – Hozzárendelési feladat	91
6.5.2 Második fázis – Kiegyenlítési feladat	99
6.6 A LIT-LMU vágási szabály	101
7. ERŐFORRÁS KIEGYENLÍTÉSI HEURISZTIKÁK	105
7.1 Prioritások alkalmazása	107
8. ÖSSZEFOGLALÁS	116
8.1 Tézisek	116
8.2 További kutatási irányok	118





## A PROJEKTMENEDZSMENT ALAPFOGALMAI

A szervezetek tevékenysége mindig valamilyen cél vagy célrendszer elérésének irányába hat. Az ehhez szükséges feladatokat, illetve a teljesítés közbeni körülményeket külső és belső adottságok határozzák meg, melyek rövid és hosszú távon is változásokat kényszerítenek a szervezetekre. A rövidtávú, főleg belső feladatokhoz kapcsolódó napi folyamatos rutinmunka, valamint a hosszútávú, többnyire külső körülmények által behatárolt változások végrehajtása között meghúzódó széles tevékenységi skála jól érzékelteti, hogy a szervezetek vezetése többdimenziós feladat. Az *operatív menedzsment* a közvetlen célok eléréséért, a folyamatos működés megteremtéséért valamint a rövidtávú eredményességért felelős. A *stratégiai menedzsment* feladata a szervezet hosszútávú céljainak kitűzése, és a remélt célállapothoz vezető úton a szervezet belső adottságainak és a külső körülményeinek szinkronban tartása. A stratégiai célok megvalósítása általában több, de egyenként jól körülhatárolható rész cél elérésére bontható, melyek időben elkülönülve párhuzamosan vagy részben átfedésekkel, szakmailag pedig több területet is átfogó ismeretekkel valósíthatóak meg. Az ilyen jellegű feladatok kivitelezése a fent említett menedzsment funkcióktól eltérő vezetési szemléletet, de főleg más technikák és módszerek alkalmazását igénylik. A *projektmenedzsment*, mint a harmadik vezetési dimenzió éppen ezt a feladatot látja el, így lényegében összekötő láncszem a hagyományos operatív és stratégiai menedzsment diszciplínák között. Feladatából fakadóan a szervezeti, tervezési, személyzeti, vezetési és kontrolling ismeretekén túl, feltételezi az operációkutatás, a matematikai és statisztikai módszerek használatát is. A jó *projektmenedzser* olyan személy, akinek nemcsak felelőssége és hatásköre van a részfeladatok teljesítéséhez, hanem ezen túlme-



nően a szükséges szakterületen is jártas. Ki kell igazodnia a munkáját támogató módszerekben és szoftverekben egyaránt. Ritkábban ugyan, de olyan tudományterületek ismeretanyagára is szüksége lehet, mint a jog, a magatartástudomány, a pszichológia vagy a logisztika.

A projektmenedzsment kifejezést azonban gyakran nem mint vezetési feladatkör egyik dimenzióját használjuk, hanem mint egy konkrét módszertani rendszert. A projektmenedzsment ebben a vonatkozásában egyszerre jelent új technológiát, új munkamódszereket, kézi és automatizált eljárásokat, vagy megváltozott emberi kapcsolatokat. A különféle szabványosított projektmenedzsment módszertanok (például PRINCE, SSADM) olyan szocio-technikai rendszerek, amelyek megvalósulását és működését főleg azon környezet szervezeti és személyzeti tényezői határozzák meg, amelyre telepítették. Ezért a projektmenedzsment hatékonysága egyaránt függ annak mind technikai, mind szervezeti vonatkozásaitól. [1.11]

Ezt a kettősséget a jelen dolgozat is magán viseli. Először az ismertetendő modellünket helyezzük el a projektmenedzsment (mint vezetési dimenzió) rendszerében, majd pedig az új elemeket felhasználva egy konkrét, alkalmazásra kész megoldási eszközt (részmodszertant) adunk közre. Az új ütemezési modell két területen is összekötő jelentőségű: Egyrészt az erőforrások módszertani kezelésénél figyelembe vesszük azok humán aspektusait is, másrészt lehetővé tesszük az időkorlátos és az erőforrás-korlátos tervezési módszerek egységes kezelését.

## 1.1 A projekt

A szervezetek működésük során (az operatív és a stratégiai menedzsment funkciókhoz kapcsolódóan) alapvetően kétféle tevékenységet végeznek: rutinmunkát, illetve projektmunkát. A két véglet között természetesen széles skála húzódik meg, sőt a rutinmunka a projektmunkákkal gyakran át is lapolódik. Mindkettőnek vannak közös jellegzetességei: (1) emberek végzik, (2) a

lehetőségeket erőforrások korlátozzák, (3) tervezési, kivitelezési és ellenőrzési fázisokon mennek keresztül.

A leglényegesebb megkülönböztető jegyük az, hogy a rutinmunka tartós, ismétlődő feladatokból és tevékenységekből, míg a projektmunka ideiglenes, és egyedi tevékenységekből áll. A projektek nincsenek sem szervezeti szinthez, sem létszámhoz kötve. Egyszemélyes projektek éppúgy léteznek, mint több ezer embert magába foglalóak. Hasonlóan végezheti egy szervezeti egység vagy több szervezeti egységet átfogó keresztmetszeti struktúra is, de akár szervezeten kívüli partnerek is közreműködhetnek benne. A projektek léte és kezelése gyakran kulcsszerepet játszik a szervezetek üzleti stratégiájában.

A fejezetben szereplő definíciók a Magyar Projektmenedzsment Szövetség szótárának ajánlásai alapján készültek.

### Definíció

*Projektnek* tekintjük azon tevékenységek és folyamatok összességét, amelyek *egyszeri komplex* tevékenységfolyamat által végrehajthatóak, egyszeri konkrétan körülhatárolt *célt* kell elérniük, jól meghatározott *költségkeretek* között, pontosan rögzített *határidőre*.

A definícióból közvetlenül következik a *meg nem ismételhetőség*, melynek értelmében minden projekt olyan egyszer elvégzendő feladat, amely egyértelműen beazonosítható. A világosan definiált cél következtében mindig valamiféle eredménye van (még kudarc esetén is), mely általában egy adott termék illetve szolgáltatás, vagy bármi egyéb, amely pontosan meghatározható. A projekteknek ezért általában egyedi megnevezéseik is vannak. [1.6]

A másik fontos következmény az *ideiglenesség*, mely alatt azt értjük, hogy a projekthez kapcsolódó változások átmeneti jellegűek. A projekteknek határozott kezdete és vége van, azaz végesek, valamint idő-, költség- és teljesítmény értékek rendelkeznek hozzájuk.

Ahhoz, hogy a rendelkezésre álló technikai eszköztárat differenciáltan és hatékonyabban lehessen alkalmazni, a projekteket - a bennük megfogalmazott célok alapján - a következő kategóriákba sorolhatjuk: (1) beruházási projektek, (2) innovációs projektek és (3) szellemi szolgáltatási projektek. [1.7]

Azokat a nagyméretű projekteket, amelyek több, különböző kategóriába tartozó alprojekteket foglalnak magukban *megaprojekteknek* nevezzük. Bár a szakirodalomban a szuperprojekt kifejezés a megaprojekt szinonimájaként használatos, a továbbiakban mégis élünk a megkülönböztetés lehetőségével, és *szuperprojekt* alatt olyan több alprojektből álló projektet értünk, ahol az alprojektek mindegyike ugyanabba a kategóriába tartozik.

*Beruházási* (vagy más néven *műszaki*) projektek esetében mindig valamilyen kézzel fogható eredmény keletkezik, ami lehet egy konkrét termék, vagy egy szolgáltatáshoz tartozó létesítmény (vagy az előbbiek megszüntetése). *Innovációs* (vagy más néven *kutatói és fejlesztési*) projekteknél a hangsúly az újdonságon van, vagyis mindig valami új termék vagy szolgáltatás készül, vagy új technológia kerül bevezetésre (esetleg új piacon), vagy létező dolgok újításáról van szó. A *szellemi szolgáltatások* körében pedig mindig új minőség jön létre, amely megjelenhet egy szervezet működési keretfeltételeinek vagy belső adottságainak megváltozásában.

A projekt céljának definiáltságát az úgynevezett SMART dimenziók mentén lehet értékelni, melynek összetevői: megkülönböztettség, mérhetőség, elfogadottság, realitás és határidő tarthatóság.

## 1.2 A projektmenedzsment

A projektek kezeléséhez jól meghatározható rendszerre van szükség, amely bizonyos ismeretek, képességek és technikák birtokában úgy képes befolyásolni az egyes tevékenységeket, hogy a projekt, mint egész megvalósítsa, vagy túlszárnyalja a projekttel kapcsolatos elvárásokat. A projekt teljesítésében



résztvevők köre alapján megkülönböztetünk külső, illetve belső projekteket. A projektekkel kapcsolatos elvárásokat a projektgazdák fogalmazzák meg.

### **Definíció**

A *projektmenedzsment* azon vezetési, irányítási folyamat, amely arra irányul, hogy a projekt eredményesen megvalósuljon, a rendelkezésre álló idő-, pénz-, és erőforráskorlátok mellett. Feladata a projektmegvalósítás folyamatának vezetése, irányítása, szervezése, amely egyrészt az erőforrásokat és az információkat, másrészt a rendelkezésre álló módszertani és technikai eszköztárat is a definiált cél elérésére összpontosítja.

A projektmenedzsment tehát egy komplex folyamat (a projekt) egészét átfogó és eredményességét segítő integrált vezetési-irányítási rendszer, amely magában foglalja a projekt egész életciklusát, a problémafeltárás, illetve koncepcióalkotás fázisától kezdve egészen a megvalósítás, az implementálás fázisáig.

A projektmenedzsment eredményessége a következő dimenziók szerint értékelhető: (1) a projekt hatóköre, ideje, költsége és minősége, (2) rögzített elvárásoknak, és remélt várakozásoknak való megfelelés.

A projektmenedzsment (mint módszertan) elemeinek rendszer szintű bemutatását az 1.1 táblázat tartalmazza. A táblázat a Project Management Institute PMBOOK módszertanának szerkezetét tekinti át. Az itt felsorolt menedzsment területek minden általános projektmegvalósítási módszertanban felismerhetőek, legfeljebb más bontásban és más hangsúlyozással. [1.1]

1.1 táblázat: A projektmenedzsmenthez kapcsolódó menedzsment területek áttekintése

Projektmenedzsment		
<p>(A) Integráció menedzsment</p> <p>jelenti azokat a folyamatokat, amelyek biztosítják a projekt elemeinek megfelelő összehangolását.</p> <p>Elemi:</p> <p>(A1) <i>Projekterv készítés</i></p> <ul style="list-style-type: none"> <li>– a többi tervezési folyamat egységesítése és dokumentálása</li> </ul> <p>(A2) <i>Projekterv végrehajtása</i></p> <ul style="list-style-type: none"> <li>– a tevékenységek terv szerinti kivitelezése</li> </ul> <p>(A3) <i>Projekterv felülvizsgálata</i></p> <ul style="list-style-type: none"> <li>– a projekt változásainak menedzselése</li> </ul>	<p>(B) Kiterjedés menedzsment</p> <p>jelöli azokat a folyamatokat, amelyek biztosítják, hogy azok és csak azok a tevékenységek kerüljenek végrehajtásra, amelyek feltétlenül szükségesek a projekt sikerességéhez.</p> <p>Elemi:</p> <p>(B1) <i>Inicializálás</i></p> <ul style="list-style-type: none"> <li>– a szervezet felkészítése a projekt során következő fázisának megkezdésére</li> </ul> <p>(B2) <i>Kiterjedés tervezés</i></p> <ul style="list-style-type: none"> <li>– a jövőbeli döntések támogatására írásos hatókör leírás</li> </ul> <p>(B3) <i>Kiterjedés definiálás</i></p> <ul style="list-style-type: none"> <li>– a hatókör kisebb, menedzselhetőbb komponensekre bontása</li> </ul> <p>(B4) <i>Kiterjedés bizonyítás</i></p> <ul style="list-style-type: none"> <li>– a hatókör feltételek formalizálása az ellenőrizhetőség érdekében</li> </ul> <p>(B5) <i>Kiterjedés felügyelet</i></p> <ul style="list-style-type: none"> <li>– a hatókör változásának kezelése</li> </ul>	<p>(C) Időmenedzsment</p> <ul style="list-style-type: none"> <li>– vizsgálja azokat a folyamatokat, melyek biztosítják, hogy a projekt időben kerüljön kivitelezésre.</li> </ul> <p>Elemi:</p> <p>(C1) <i>Tevékenység definiálás</i></p> <ul style="list-style-type: none"> <li>– a projekt kivitelezéséhez szükséges tevékenységek beazonosítása</li> </ul> <p>(C2) <i>Függőség meghatározás</i></p> <ul style="list-style-type: none"> <li>– a tevékenységek közötti precedenciák dokumentálása</li> </ul> <p>(C3) <i>Időigény becslés</i></p> <ul style="list-style-type: none"> <li>– az egyes tevékenységek elvégzéséhez szükséges periódushossz kalkulálása</li> </ul> <p>(C4) <i>Ütemezés</i></p> <ul style="list-style-type: none"> <li>– a tevékenységek összefüggéseinek elemzése, sorrendiség eldöntése az idő- és erőforrásigény figyelembevételével</li> </ul> <p>(C5) <i>Ütemezés felügyelet</i></p> <ul style="list-style-type: none"> <li>– az ütemezés elcsúszásának menedzselése.</li> </ul>
<p>(D) Költség menedzsment</p> <ul style="list-style-type: none"> <li>– fogja össze azokat a folyamatokat, amelyek biztosítják, hogy a projekt kivitelezése az adott költségvetési kereteken belül történjen meg.</li> </ul> <p>Elemi:</p> <p>(D1) <i>Erőforrás-tervezés</i></p> <ul style="list-style-type: none"> <li>– a tevékenységek végrehajtásához szükséges (eszköz, ember, anyag) mennyiségek megállapítása</li> </ul> <p>(D2) <i>Költségbecslés</i></p> <ul style="list-style-type: none"> <li>– a szükséges erőforrások pénzügyi egyenértékének közelítő kalkulációja,</li> </ul> <p>(D3) <i>Költségtervezés</i></p> <ul style="list-style-type: none"> <li>– a teljes projekt-költségvetés allokációja</li> </ul> <p>(D4) <i>Költségterv felügyelet</i></p> <ul style="list-style-type: none"> <li>– a költségek változásának menedzselése</li> </ul>	<p>(E) Minőség menedzsment</p> <ul style="list-style-type: none"> <li>– kezeli azokat a folyamatokat, amelyek az elvárásoknak való megfelelést biztosítják</li> </ul> <p>Elemi:</p> <p>(E1) <i>Minőségtervezés</i></p> <ul style="list-style-type: none"> <li>– annak meghatározása, hogy mely minőségügyi szabványok illeszthetők a projektekhez, és hogyan lehet ezeknek megfelelni</li> </ul> <p>(E2) <i>Minőségbiztosítás</i></p> <ul style="list-style-type: none"> <li>– az egyes fázisok megbízhatósági tartományának rögzítése, hogy a projekt egésze teljesítse az előírásokat</li> </ul> <p>(E3) <i>Minőségellenőrzés</i></p> <ul style="list-style-type: none"> <li>– a megbízhatósági tartományoknak való megfelelés ellenőrzése, valamint a nem kívánatos folyamatok kiküszöbölése és javítása</li> </ul>	<p>(F) Emberi erőforrás menedzsment</p> <ul style="list-style-type: none"> <li>– tartalmazza azokat a folyamatokat, amelyek a projektben résztvevők munkájának hatékonyságát hivatottak növelni.</li> </ul> <p>Elemi:</p> <p>(F1) <i>Szervezet-tervezés</i></p> <ul style="list-style-type: none"> <li>– a projekthez kapcsolódó szerepek azonosítása, hatáskör és felelősségi körök meghatározása, kapcsolatrendszer dokumentálása</li> </ul> <p>(F2) <i>Személyzettel való ellátás</i></p> <ul style="list-style-type: none"> <li>– a projekt személyzet kialakítása, vezetési alapelvek kialakítása</li> </ul> <p>(F3) <i>Csapatmunka támogatás</i></p> <ul style="list-style-type: none"> <li>– csapatépítés, az egyéni és csapat képességek és ismeretek fejlesztése a hatékonyság érdekében</li> </ul>

(G) Kommunikáció menedzsment	(H) Kockázat menedzsment	(I) Szerződés menedzsment
<ul style="list-style-type: none"> <li>– jelenti azon folyamatok összességét, amelyek az információk hatékony beszerzését, elosztását, tárolását, és a hivatali útvonalak kialakítását támogatják.</li> </ul>	<ul style="list-style-type: none"> <li>– folyamatai felelősek a projekttel kapcsolatos kockázati tényezők beazonosításáért, elemzéséért és csökkentéséért.</li> </ul>	<ul style="list-style-type: none"> <li>– fogja össze azokat a folyamatokat, amelyek biztosítják, a projekthez szükséges külső eszközök és szolgáltatások megfelelő helyről és megfelelő időpontban történő beszerzését.</li> </ul>
<p>Elemek:</p> <p>(G1) <b>Kommunikáció tervezés</b></p> <ul style="list-style-type: none"> <li>– a projekt tulajdonosok információs és kommunikációs igényeinek feltárása, kinek, mikor, kitől és milyen információra van szüksége</li> </ul> <p>(G2) <b>Információ megosztás</b></p> <ul style="list-style-type: none"> <li>– a projekt tagok és tulajdonosok közötti rendszeres információcseré kialakítása a hatékonyság növelése érdekében</li> </ul> <p>(G3) <b>Teljesítmény jelentés tervezés</b></p> <ul style="list-style-type: none"> <li>– a helyzet-megállapításhoz szükséges információk összegyűjtése és terítése, helyzetjelentési forgatókönyvek, mérési rendszer és előrejelzési módszerek kialakítása</li> </ul> <p>(G4) <b>Adminisztráció</b></p> <ul style="list-style-type: none"> <li>– a projektfázisok lezárásának dokumentálása</li> </ul>	<p>Elemek:</p> <p>(H1) <b>Kockázati források beazonosítása</b></p> <ul style="list-style-type: none"> <li>– a projektet valószínűsíthetően fenyegető tényezők összegyűjtése és kategorizálása</li> </ul> <p>(H2) <b>Kockázat mérés</b></p> <ul style="list-style-type: none"> <li>– kockázatelemzés, hatástanulmányok készítése, maradványkockázat meghatározása</li> </ul> <p>(H3) <b>Kockázati terv készítése</b></p> <ul style="list-style-type: none"> <li>– a kockázatok csökkentési módjainak meghatározása, fenyegetésekre való reakciók forgatókönyveinek kidolgozása</li> </ul> <p>(H4) <b>Kockázat felügyelet</b></p> <ul style="list-style-type: none"> <li>– a projekt kockázatában bekövetkező változások menedzselése</li> </ul>	<p>Elemek:</p> <p>(I1) <b>Szükséglet-tervezés</b></p> <ul style="list-style-type: none"> <li>– mikorra és mit kell biztosítani a projekt részére</li> </ul> <p>(I2) <b>Beszerzés-tervezés</b></p> <ul style="list-style-type: none"> <li>– a szükségletre vonatkozó követelmények dokumentálása,</li> <li>– a lehetséges beszerzési források számbavétele</li> </ul> <p>(I3) <b>Szelekció</b></p> <ul style="list-style-type: none"> <li>– mennyiségi és árajánlatok begyűjtése, elemzése, alku</li> </ul> <p>(I4) <b>Kiválasztás</b></p> <ul style="list-style-type: none"> <li>– a legmegfelelőbb potenciális beszállítók kijelölése,</li> </ul> <p>(I5) <b>Szerződés adminisztráció</b></p> <ul style="list-style-type: none"> <li>– szerződéskötések és partnerkövetés</li> </ul> <p>(I6) <b>Szerződés lezárás</b></p> <ul style="list-style-type: none"> <li>– a szerződések lezárása, értékelése</li> </ul>

A projektmenedzsment stratégiai elemei a projekt céljainak hatékony és eredményes realizálását célozzák, melynek leglényegesebb tényezője a *rugalmasság* és *adaptivitás* megőrzése. Ennek érdekében a projektmenedzsment feladata megvizsgálni a projektre ható tényezőket, melyeket az alábbi kategóriákba lehet sorolni: (a) környezeti hatások, (b) a szervezeti hatások és (c) hajtó- illetve fékezőerők.

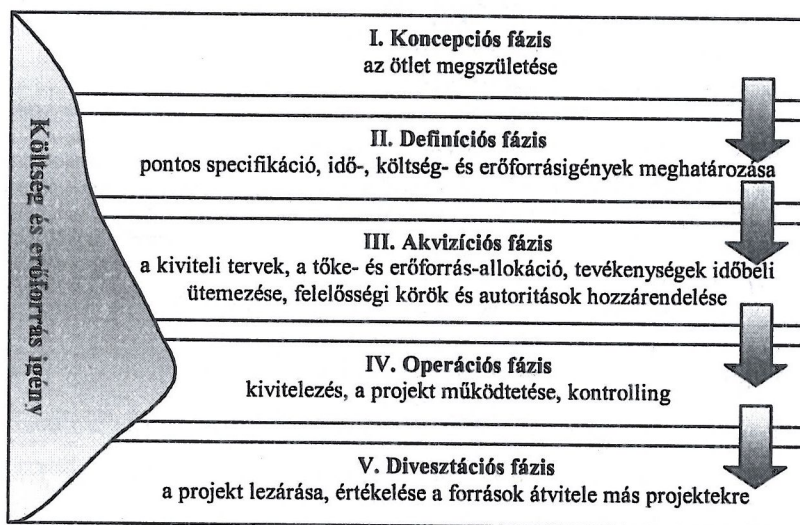
### 1.3 A projekt életciklusa

A projektmenedzsment feladatai időben éppúgy behatároltak, mint magának a projektnek a kivitelezése. A projekt megszületésétől annak lezárásáig tartó folyamat egymástól jól elhatárolható koncepcionális fázisokra bontható. Az egyes fázisok közötti kapcsolatrendszer ábrázolására háromféle módszer terjedt el: (1) a lineáris, (2) a hierarchikus és (3) a spirális.



A lineáris ábrázolásmód (1.1 ábra) esetében az egyes fázisok közötti kapcsolatrendszer meghatározása az időbeni egymásutánosság alapján történik. A hierarchikus szemlélet szerint a kapcsolatok a fázisok közötti logikai összefüggéseket tükrözik. A spirális esetben (3.2 ábra) mindig ugyanazon fázisok körforrás szerinti ismétlésére kerül sor, csak egyre magasabb szinten.

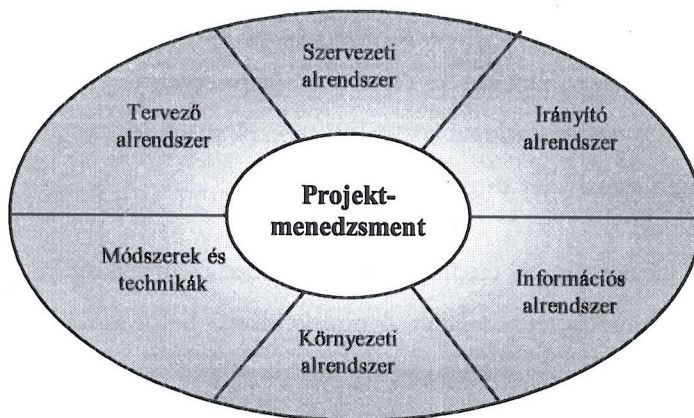
A projekt céljának, folyamatainak és szervezetének összehangolását az általános projektmenedzsment szintjén az alábbi lineáris projekt-életciklus modellel lehet leírni [1.3]:



1.1 ábra: A projektmenedzsment életciklusa

A projektmenedzsment, mint vezetési gyakorlat, leggyakrabban rendszerelvű („holisztikus”, vagy „fuzzy-elvű”) közelítést képvisel azáltal, hogy egyik irányban sem elfogult, hanem egyidejűleg képviseli a vállalat valamennyi érintett szakterületének érdekeit. Így a projektmenedzsment egyik legfontosabb funkciója a különböző szakterületek, illetve szervezeti egységek magas színvonalú koordinációjának a biztosítása. [1.5]

A projektmenedzsment rendszerszemléletű megközelítésénél az alábbi elsődleges alrendszereket azonosíthatjuk:



1.2 ábra: A projektmenedzsment rendszerorientált megközelítése

Az eddig ismertetett logikai rendszerben elhelyezve a bevezetésre kerülő ütemezési modell a tervezési alrendszerhez, az életciklus akvizíciós fázisához, valamint a projektmenedzsment módszertanok időmenedzsment és emberi erőforrás menedzsment területéhez kapcsolódik.

#### 1.4 Projektek sikertényezői

A projektmenedzsmenttel foglalkozó kutatások egyik legkiábrándítóbb megállapítása az, hogy a projektek sikeres kivitelezéséhez nem adható meg örökérvényű, egységes szabály, még azonos típusú projektek esetében sem. A projektek nagyfokú változékonysága gyakran még egy adott kategórián belül is lehetetlenné teszi a sikeresség kritikus tényezőinek pontos meghatározását. [1.2]

Mitől is függ a projekt és/vagy a projektmenedzsment sikeressége? Nem az a lényeges, hogy milyen eszközöket és módszereket használunk fel és milyen mértékben, hanem sokkal inkább az, hogy azok mennyire felelnek meg a konkrét feladatoknak és alkalmazási feltételeknek. A sikeres projekt igen sok tényező együttes eredményeként alakul ki. A projekt sikerét a definícióban is szereplő három dimenzió mentén definiálhatjuk:

## Definíció

*Sikeres* az a projekt, amelyet az (1) eredeti ütemterven és (2) költségvetésen belül és az (3) eredeti szakmai követelményeknek megfelelő, vagy annál magasabb szinten valósítanak meg.

Az előbb idézett negatív eredmény ellenére érdemes megvizsgálni néhány ide vonatkozó felmérés tapasztalatait. A projekt sikerét befolyásoló tényezők relatív fontosságuk szerinti sorrendje [1.8] szerint például a következőképpen adható meg: (1) Projekt misszió, (2) Ütemterv, (3) Konzultáció a projekt gazdái-val, (4) Technikai feladatok, (5) Elfogadottság, (6) Felügyelet és visszacsatolás, (7) Kommunikáció, (8) Konfliktus és hibaelhárítás, (9) Emberi erőforrások (toborzás, kiválasztás, képzés) és (10) Menedzsment támogatása.

Láthatjuk, hogy a sikertényezők zöme a menedzsment tudományokhoz kötődik. A technikai sikertényezők közül pedig az ütemterv készítésének problémaköre került a második helyre. Noha ezeket a következtetéseket egyetlen felmérés alapján vontuk le, mégis jogosan várhatjuk azt, hogy egy olyan ütemezés-tervezési módszer, amely alkalmas néhány menedzsmenthez kötődő sikertényező kezelésére is, nagymértékben hozzájárulhat a projektek sikerességének javításához.

A továbbiakban az 1.1 táblázat emberi erőforrás (F) és időmenedzsment (C) oszlopait fejtjük ki részletesebben. Érdemes megemlíteni azonban, hogy például a dolgozat magját képező ütemezési modellnek két olyan változata is ismeretes, amelyek közül az egyik a H. kockázat menedzsment [1.10], a másik pedig a D. pénzügyi tervezés [1.9] témakörében nyújt segítséget.



## EMBERI ERŐFORRÁS ÉS PROJEKTMENEDZSMENT

Minden projekt a maga meg nem ismételtetősége és ideiglenes jellege ellenére hosszú távon befolyással bír a projekt tulajdonos szervezetre. Minden egyes projekteredmény – függetlenül annak sikerességétől vagy sikertelenségétől – hozzájárul a jövőbeni projektek bizonytalanságából eredő kockázati tényezők csökkentéséhez. A *pilot projekt* módszer éppen az ilyen jellegű építő hatás közvetlen kiaknázásán alapul.

A szerzett tapasztalatokon túl a kipróbált vagy alkalmazott új technológiák és módszerek is hozzájárulnak a későbbi feladatok pontosabb felméréséhez és megoldásához. Gyakran maga a projekt is innovációként honosul meg a szervezetben, és így kiváló alkalmat nyújt mind az egyéni, mind a szervezeti tanulás számára. A felhasznált új technológiák jó része azonban nem jelent automatikusan új tudást a szervezet számára. Ennek megfelelően „az innováció a helyes új megközelítés” elvének alkalmazása tudatos tervezést igényel. [2.1]

A projektmenedzsment eszköztárában a tapasztalatokból származó ismeretek rögzítésére használatos eszköz a projektzáró jelentés. Ebben a dokumentumban mind a projektmenedzsment saját szakmai tapasztalatai, mind pedig magával a projekt tárgyával kapcsolatos szakmai tapasztalatok bekerülhetnek. A következő fejezetben részletezendő informatikai projektek esetében ez utóbbi jórészt automatikus. A technológiai tapasztalatok, az alkalmazott módszertan, az elkészült és újra felhasználható szoftvermodulok megfelelő dokumentálás után a szervezet egésze számára elérhetőek lesznek. Éppen ezért kell (főleg szellemi töke-igényes iparágakban) nagyobb gondot fordítani projektmenedzsment tapasztalatok kvantitatív és kvalitatív adatainak újrahasznosíthatóságára.

Előrebocsátva, hogy az informatikai projektek fő jellegzetessége a nagy szellemi töltetű igény, ezért az 1.1 táblázat emberi erőforrás menedzsment blokkjának struktúráját egy szinttel mélyebb bontásban is bemutatjuk.

2.1 táblázat: A projektmenedzsment emberi erőforrásokhoz kapcsolódó funkciói

Emberi erőforrásokhoz kapcsolódó menedzsment funkciók projektekben		
<p>(F1) Szervezet tervezés</p> <p>szerep- és felelősségi körök valamint függőségek azonosítása, dokumentálása, külső és belső szervezetek összekapcsolása, folyamattervezés</p> <p><b>Inputok:</b></p> <p>(F1-I1) <i>Projekt interfészek</i></p> <ul style="list-style-type: none"> <li>- a szervezeti egységek közötti formális és informális interfészek</li> <li>- a technikai területekhez kapcsolódó formális és informális interfészek</li> <li>- interperszonális interfészek a projekt tagjai között</li> </ul> <p>(F1-I2) <i>Személyzeti igény</i></p> <ul style="list-style-type: none"> <li>- milyen alkalmazottaktól, milyen képességekre, milyen időtartamra van igény</li> </ul> <p>(F1-I3) <i>Korlátozó tényezők</i></p> <ul style="list-style-type: none"> <li>- a projektgárda összeállításának korlátozó tényezői (a befogadó szervezet korlátai, kollektív szerződések megkötései, újraképzett projektgárda előzetes preferenciái, a projekttagok képzettsége)</li> </ul> <p><b>Eszközök és módszerek:</b></p> <p>(F1-E1) <i>Sablonok</i></p> <ul style="list-style-type: none"> <li>- eddigi belső, vagy más külső projektek újratervezhető részei</li> </ul> <p>(F1-E2) <i>Emberi erőforrás gyakorlat</i></p> <ul style="list-style-type: none"> <li>- bejáratott szerepek és szereposztások alkalmazása</li> </ul> <p>(F1-E3) <i>Szervezet elmélet</i></p> <ul style="list-style-type: none"> <li>- annak is kimondottan a projekt szervezetekkel foglalkozó részének ajánlásai</li> </ul> <p>(F1-E4) <i>Stakeholder analízis</i></p> <ul style="list-style-type: none"> <li>- a projekttel kapcsolatos elvárások feltérképezése</li> </ul> <p><b>Outputok:</b></p> <p>(F1-O1) <i>Szerepköri és hatáskör leírások</i></p> <ul style="list-style-type: none"> <li>- hatalmi és felelősségi tartományok</li> </ul> <p>(F1-O2) <i>Munkaidő beosztási terv</i></p> <ul style="list-style-type: none"> <li>- időskálán feltüntetett erőforrás felhasználási diagramok</li> </ul> <p>(F1-O3) <i>Szervezeti ábra</i></p> <ul style="list-style-type: none"> <li>- a projektszervezet főbb elemeinek és azok függőségének ábrázolása</li> </ul>	<p>(F2) Személyzettel való ellátás</p> <p>a projekt céljának legmegfelelőbb személyzet kiválasztása, a megfelelő képzettségű egyének és csapatok kialakítása</p> <p><b>Inputok:</b></p> <p>(F2-I1) <i>Személyzeti menedzsment tervek</i></p> <ul style="list-style-type: none"> <li>- hogyan történik a projektbe illesztés illetve kivétel, mik a várható mozgások</li> </ul> <p>(F2-I2) <i>Személyzeti profilok</i></p> <ul style="list-style-type: none"> <li>- az emberek előzetes munkatapasztalatai, személyi érdeklődése, elkötelezettsége, a csapatok munkaképessége, hatékonysága és elérhetősége</li> </ul> <p>(F2-I3) <i>Toborzási praktikák</i></p> <ul style="list-style-type: none"> <li>- kiválasztási politika, munkaköri szabályzatok, folyamatszabályozások, melyek korlátozzák a csapatkialakítás menetét</li> </ul> <p><b>Eszközök és módszerek:</b></p> <p>(F2-E1) <i>Tárgyalás, közvetítés</i></p> <ul style="list-style-type: none"> <li>- a koordinátorok biztosítása arról, hogy a projekt időben megkapja a szükséges erőforrásokat, jutalmazási rendszer kidolgozása</li> </ul> <p>(F2-E2) <i>Előzetes kijelölés</i></p> <ul style="list-style-type: none"> <li>- amennyiben a projektben való részvételre alul- vagy felülméretett versenyhelyzet alakul ki</li> </ul> <p>(F2-E3) <i>Személyzet insourcing</i></p> <ul style="list-style-type: none"> <li>- a szervezeten belül nem megoldható erőforrások felkutatása és a projektbe illesztése</li> </ul> <p><b>Outputok:</b></p> <p>(F2-O1) <i>Feladat-kiosztási lista</i></p> <ul style="list-style-type: none"> <li>- a kijelölt személyzet feladatainak és alkalmazási státuszának dokumentálása</li> </ul> <p>(F2-O2) <i>Projektszemélyzet lista</i></p> <ul style="list-style-type: none"> <li>- a stakeholdereket is magában foglaló teljes csapattípus</li> </ul>	<p>(F3) Csapatmunka támogatás</p> <p>A személyzet és a stakeholderek egyéni (technikai és menedzsment) fejlesztése, csapatépítés</p> <p><b>Inputok:</b></p> <p>(F3-I1) <i>Projektszemélyzet lista</i></p> <ul style="list-style-type: none"> <li>- amely impliciten tartalmazza a felhasználható egyéni és csapat-képességeket</li> </ul> <p>(F3-I2) <i>Projekt terv</i></p> <ul style="list-style-type: none"> <li>- a projekt technikai környezete</li> </ul> <p>(F3-I3) <i>Munkaidő beosztási terv</i></p> <ul style="list-style-type: none"> <li>- részletesen lásd F1-O2</li> </ul> <p>(F3-I4) <i>Teljesítménymérési terv</i></p> <ul style="list-style-type: none"> <li>- mérési elvek és szabályok, a projekt tervhez képesti haladás visszajelzése</li> </ul> <p>(F3-I5) <i>Külső visszacsatolás</i></p> <ul style="list-style-type: none"> <li>- a projekt tagokat időről-időre szembesíteni kell a projekttel szembeni elvárásokkal</li> </ul> <p><b>Eszközök és módszerek:</b></p> <p>(F3-E1) <i>Csapatépítés</i></p> <ul style="list-style-type: none"> <li>- tulajdonképpen ide sorolható az összes általános csapatépítési eszközrendszer (elkötelezés, delegálás, konfliktuskezelés stb.)</li> </ul> <p>(F3-E2) <i>Menedzsment képességek</i></p> <ul style="list-style-type: none"> <li>- általános menedzsment ismeretek és eszközök, melyek a hatékonyság és a megelégedettség szintjét emelik</li> </ul> <p>(F3-E3) <i>Ösztönző rendszer</i></p> <ul style="list-style-type: none"> <li>- mérési, követési és díjazási rendszerek kidolgozása</li> </ul> <p>(F3-E4) <i>Kollokáció</i></p> <ul style="list-style-type: none"> <li>- a kulcsemberek térbeni közös elhelyezése</li> </ul> <p>(F3-E5) <i>Tréning</i></p> <ul style="list-style-type: none"> <li>- tudás és képességfejlesztés</li> </ul> <p><b>Outputok:</b></p> <p>(F3-O1) <i>Hatékonyságfejlesztési terv</i></p> <ul style="list-style-type: none"> <li>- egyéni és csapatszintű képzésterv</li> </ul> <p>(F3-O2) <i>Teljesítménymérési kézikönyv</i></p> <ul style="list-style-type: none"> <li>- a projekt összes tagja számára</li> </ul>



## 2.1 Team- és projektszervezetek

Azt, hogy a PMBOOK módszertan a 2.1 táblázat szerint az (F1) szervezet-tervezést kissé szokatlan módon az emberi erőforrás menedzsment részeként tünteti fel, most ki is használjuk arra, hogy röviden áttekintsük a projektszervezetek témakörét. A projektmenedzsment területén alkalmazott leggyakoribb strukturális típusú koordinációs eszköz a projektgárda\* és a team. Kisebb arányban, de találkozhatunk a többi strukturális megvalósítással (bizottságok, törzskarok és termékmenedzseri rendszer) is.

### Definíció

A *projektgárdában* különböző motivációjú, eltérő ismeretekkel és képességekkel rendelkező, a szervezetek különböző szakmai területein és hierarchikus szintjein elhelyezkedő emberekből képezünk új, ideiglenes mikroszervezeteket.

### Definíció

A *team* a szervezet különböző területein tevékenykedő, eltérő pozíciókkal rendelkező személyekből álló feladatorientált és autonóm egység, amelyet valamilyen probléma megoldására, illetve ideiglenes vagy állandó feladat elvégzésére hoznak létre.

Ezt a kétféle mikroszervezetet egy adott (nem projekthez kötődő) cél elérése érdekében is érdemes létrehozni, hiszen (1) a projekt esetében átlátható az időtáv, adott a költségkeret, a munka problémaorientált, valamint (2) a vegyes szakmai összetétel támogatja a részérdekek összehangolását, továbbá (3) a vállalati struktúra átalakítása nélkül is kivitelezhető.

A projektcsoporthierarchikusan strukturált vagy team formájában működő szervezeti egység, amelynek vezetése egyértelmű kompetenciával és felelősséggel rendelkezik. A projektgárda tagjai – a projekt fontosságától függően –

---

\* A projektgárda kifejezés a szakirodalomban nem használatos. A szervezeti formák elméletében elfogadott megnevezése a projekt, amely szó jelen tárgyalásmód mellett sajnos már foglalt.

az eredeti szervezeti egységeiből kikerülve függelmileg és szakmailag is a projektvezető alá tartoznak [2.14].

A különbség a projektgárdához képest abban ragadható meg, hogy itt a probléma nem feltétlenül újszerű (*egyszeriség*), valamint időbeli korlátozottság (*ideiglenesség*) sem feltétlenül jellemző rá. Kissé leegyszerűsítve a projekteket gyakran csapat formában működtetik, amelyet *projekt-teamnek* nevezünk.

A projekt definíciójában szereplő egyszeriségből és ideiglenességből közvetlenül adódik, hogy a hozzá tartozó szervezeti formát a duális szervezeti alapprofánák között kell keresnünk. A projektek sikeres megvalósításának legtöbbször a team- és projektszervezet jellegű megoldások felelnek meg.

### **Definíció**

A *team-szervezet* olyan szervezeti forma, amely esetében az alapstruktúra (elsődleges vagy primer szervezet) jelentős megváltoztatása nélkül teamek egész rendszere, mintegy hálózatként ráépül a szervezetre.

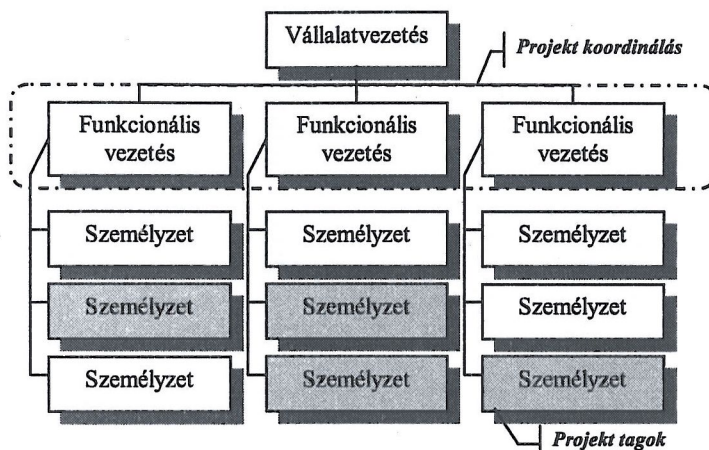
Ez a másodlagos ráépülés biztosítja, hogy a team-szervezet könnyen és gyorsan képes a környezeti feltételekhez alkalmazkodni, valamint a célul kitűzött feladatokra koncentrálni.

### **Definíció**

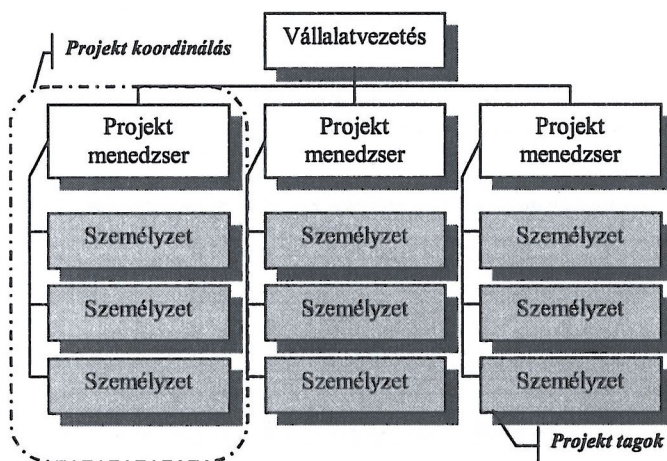
A *projektszervezet* olyan szervezeti forma, amelynek jövedelme jelentős mértékben projektekből származik, ezért a projektekhez rendelt megfelelő ideiglenes szervezeti formák a meghatározóak, melynek következtében az alapstruktúra válik másodlagossá.

A projektszervezetek legjellemzőbb ismérve a „projektek alapján történő vezetés”, azaz rendelkeznek több párhuzamosan futó projekt elszámolására, követésére, összehangolására és riportálására szolgáló menedzsment rendszerrel. Az alapstruktúra jellegének erőssége illetve elhalványulása szerint ötféle projektszervezetet különböztetünk meg.





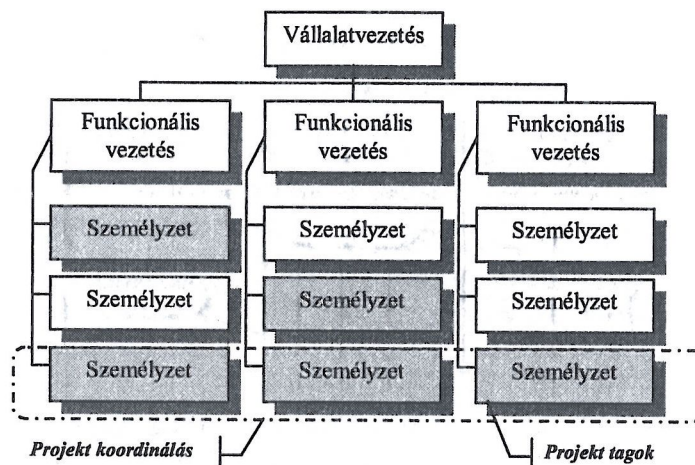
2.1 ábra: Funkcionális szervezeti struktúra



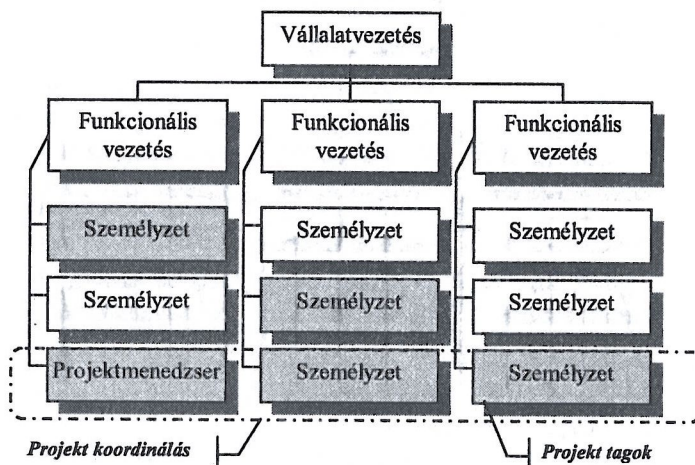
2.2 ábra: Projektesített szervezeti struktúra

1. Funkcionális szervezetekben a projektmenedzser sokkal inkább hasonlít egy információs központra, mint a projektfolyamatok valós koordinátorára és integrátorára.
2. Projektesített szervezeteknél a projektmenedzser hatásköre és felelőssége egészen a projekt határáig terjed, szerepe egyértelműen döntéshozó jelleget ölt.

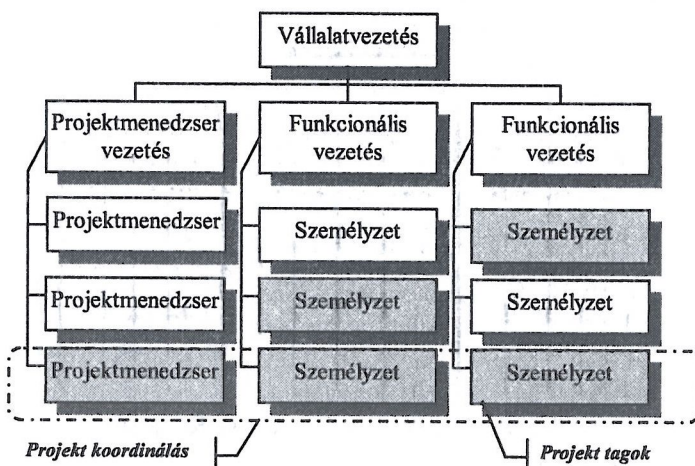




2.3 ábra: Gyenge mátrix szervezeti struktúra



2.4 ábra: Kiegyensúlyozott mátrix szervezeti struktúra



2.5 ábra: Erős mátrix szervezeti struktúra

3-4. Mátrix szervezeteknél, azok projekt orientáltsági szintje szerint három esetet különböztethetünk meg aszerint, hogy a funkcionális és a projektmenedzserek közötti kompromisszumok kialakításakor melyik fél szerepe a domináns. Gyenge mátrix szervezetek esetében a funkcionális menedzser, míg erős mátrix szervezetek esetében a projektmenedzser vezetői hatásköre az erősebb.

5. Kiegyensúlyozott mátrix esetén – mint azt a neve is mutatja –, a vezetők döntési kompetenciája egyenlő, vagy egyenlő hatásköri területekre osztott.

Az 2.2 táblázatban összefoglaljuk, hogy a projektet befogadó szervezet milyen hatással van a projekt egyes szereplőire, valamint karakterisztikáira [2.15].

2.2 táblázat: A befogadó szervezet struktúrájának hatása a projektre

<b>Projekt tulajdonság</b>	<b>Szervezeti típus</b>	<b>Funkcionális</b>	<b>Gyenge mátrix</b>	<b>Kiegyensúlyozott mátrix</b>	<b>Erős mátrix</b>	<b>Projekttest</b>
<b>Projektvezető hatalma</b>		alig van	korlátozott	közepes	Magas	Teljes
<b>A teljes munkaidős projekt tagok száma a szervezetben</b>		0%-5%	0%-25%	15%-60%	50%-95%	85%-100%
<b>Projektmenedzser státusza</b>		részidős	részidős	teljes idős	teljes idős	teljes idős
<b>Projektvezető szerepének megnevezése</b>		Projekt koordinátor / Projektvezető	Projekt koordinátor / Projektvezető	Projektmenedzser / Projektfelelős	Projektmenedzser / Program menedzser	Projektmenedzser / Program menedzser
<b>Projektmenedzsment adminisztratív csapata</b>		részidős	részidős	részidős	teljes idős	teljes idős

## 2.2 Az emberi erőforrás menedzsment és az ütemezés kapcsolatrendszere

A projektekkel kapcsolatos emberi erőforrás menedzsment kérdését sokáig az idő és tevékenység menedzsment eszközrendszerétől elkülönítve vizsgálták. A menedzsment oldaláról olyan kérdésekre keresték a választ, mint például melyek a projektek sikerességének vagy bukásának kulcstényezői, milyen szervezeti formák felelnek meg leginkább ezeknek a feladatoknak, melyek a projektvezetők kívánatos vezetői tulajdonságai [2.2, 2.3], a csoportmunka hogyan



segíti elő a projektek kivitelezésének hatékonyságát, vagy hogyan integrálható a szervezeti tanulás a projekt kivitelezésének folyamatába. Nemzetközi projektek esetén felmerül a kultúrák közötti eltérések kezelésének problémaköre is, amely az Internet elterjedésével a szoftverfejlesztésben különösen fontos szerepet kap. [2.4]

Az emberi erőforrás menedzsment és az időmenedzsment módszereinek közelítése, amelyben a magatartástudomány ismeretanyagát használják fel a tevékenységek idő és erőforrásigényének pontosabb megbecsléséhez – nem túl régi keletű. [2.5] Ugyanezen a területen jelentek meg közös eredmények a korlátozás-elmélet kapcsán a kritikus láncok módszeréhez kapcsolódva is. [2.8] A kritikus láncok módszere lényegében a kritikus út (lásd 4.4 fejezet) módszerének olyan kiterjesztése, amely egyetlen erőforrást képes kezelni. [2.10] Mégis azt mondhatjuk, hogy az emberi erőforrás menedzsment és a projektmenedzsment technikák kutatásai lényegében mind a mai napig külön szálon futnak.

A 2.1 táblázat további két (F2 és F3) oszlopának tárgyalását teljes egészében a Görög-Ternyik [2.16] irodalom „Erőforrás-tervezés” című fejezetéből kölcsönöztük. A részeiben aktualizált, de lényegében szó szerinti felhasználás oka egyrészt az, hogy a magyar nyelvű szakirodalomban ez a témakörhöz legszorosabban kapcsolódó olyan jellegű munka, ahol az idő- és erőforrás-tervezés kölcsönhatásai megjelennek. Másrészt éppen az idézendő fejezetrész adta a modell kialakításának alapötletét.

## **2.3 Erőforrás-tervezés**

A projekt időterve és a teljesítéshez igénybe vett erőforrások között kölcsönös összefüggés áll fenn. Egy adott tevékenység időtartama (és így végső soron a projekt teljesítésének határideje is) alapvetően a benne foglalt munkamennyiségtől és bizonyos külső korlátozó körülményektől (például részhatáridők) függ. Ezen túl a tevékenység időtartama leginkább a hozzá rendelt erőforrások

mennyiségére és azok teljesítőképességére érzékeny. A teljesítés időtartama, valamint a teljesítéshez igénybe vett erőforrások mennyisége és teljesítőképessége bizonyos határok között átváltható. A projekt időterve előtt ezért meg kell vizsgálnunk a teljesítéshez szükséges erőforrásokat és azok tulajdonságait is.

Az erőforrások - különösen a materiális jellegű erőforrások – rendkívül heterogének lehetnek, így a különféle projektek teljesítésének erőforrás-tervezése tevékenység specifikus megközelítést igényel. Minden projektnek szüksége van ugyanakkor egy többé-kevésbé homogén erőforrásra, nevezetesen: az emberi erőforrásra. A tervezéshez két, a gyakorlatban is igen jól alkalmazható eszköz áll rendelkezésre, mégpedig: (1) az emberi erőforrások szakmai leltárának mátrixa, valamint (2) a feladat/felelősség mátrix.

2.3 táblázat: Az emberi erőforrások szakmai leltárának mátrixa

Megnevezések	Modellezési ismeretek	Programnyelv ismeret	Fejlesztői eszköztár	Minőség-biztosítás
Kiss J.	UML	Java - magas	JUnit	
Kovács Z.		Java - közép	NetBeans	
Tóth S.	SSADM	PHP		
Nagy N.			JUnit	Validáció

Az emberi erőforrások szakmai leltára mátrixának bal oldalán, oszlopszerűen, egymás alatti felsorolásban helyezkednek el a rendelkezésre álló erőforrások, a felső vízszintes sorban pedig a projekt teljesítéséhez szükséges szakmai kvalitások találhatók. Az így kialakuló mátrix négyzeteiben jelölhető az, hogy kik, milyen (szükséges) szakmai ismeretekkel és képességekkel rendelkeznek.

2.4 táblázat: A feladat/felelősség mátrix

Megnevezések	Kiss J.	Kovács Z.	Tóth S.	Nagy N.
Design - technikai specifikáció elkészítése	1 hét (100%) Felelős	1 hét (20%) Közreműködő		
Programozás – kódolás, modul-fejlesztés		2 hét (80%) Felelős		
Tesztelés – verifikáció és validáció				1 hét (100%) Felelős
Dokumentálás			1 hét (100%) Felelős	

A feladat/felelősség mátrix a bal oldalon egymás alatti felsorolásban tartalmazza a teljesítendő tevékenységeket, a felső vízszintes sor pedig a választott emberi erőforrásokat. Az így létrejövő mátrix négyzeteiben ugyancsak jelölhető az, hogy az egyes feladatok teljesítéséért ki, (esetleg milyen vonatkozásban) tartozik felelősséggel és közreműködéssel (végrehajtás, döntés, stb.). Ugyancsak itt jelölhető az időtartam is, szükség esetén akár dátumszerűen.

Egy projekt erőforrástervének kialakítása során a formai elrendezés tekintetében a feladat/felelősség mátrixhoz hasonlóan összefoglalható az egyéb erőforrások iránti igény is. Ekkor a mátrix felső vízszintes sorában a projektben résztvevő munkatársak helyett a szükséges anyagok, berendezések, eszközök, szoftverek, stb. megnevezése kerül, míg a bal oldali oszlop változatlanul az időterv tevékenységeit tartalmazza. A mátrix négyzeteiben az időtartamra vonatkozó adatok mellett jelölhető a mennyiségre, a kapacitásra stb. utaló minden lényeges információ is.

Ismételten az időtartamra és az erőforrások közötti kölcsönösségre hivatkozva említést érdemelnek az erőforrás-tervezésben kialakult - részben számítógépes programok hatására is elterjedő - szemléletmódok.

Ezek egyike az úgynevezett *időkorlátos erőforrás-tervezés*, a másik pedig az úgynevezett *erőforrás-korlátos tervezés*. Az első azt feltételezi, hogy a projekt teljesítési időtartama ugyan kötött, de az erőforrások korlátlanul rendelkezésre állnak. A második viszont abból indul ki, hogy az erőforrások korlátozottak, következésképpen a teljesítési időtartamot alá kell rendelni a rendelkezésre álló erőforrásoknak. A gondolatmenet logikája egyik esetben sem állja meg a helyét, még tisztán belső projektek esetében sem. *(Itt jegyezzük meg, hogy a modell egyik leglényegesebb újonsága éppen abban rejlik, hogy ezt a kétféle megközelítést egyszerre képes kezelni.)* Ritka kivételként fordul elő, hogy a projektteljesítéshez szükséges összes erőforrás korlátlanul rendelkezésre álljon. Ha csak az erőforrások egyetlen eleme is szűk keresztmetszetet okoz, máris zsákutcába jutott a teljesítés. A



teljesítés időtartamának lényeges megnövelése pedig kétségesse teszi a stratégiai célok elérését.

Valójában e szélsőséges szituációkat feltételező gondolatmenet helyett célszerűbb, ha az időtartamok és az erőforrások közötti ellentétet az alábbi kompromisszumok szerint próbáljuk feloldani: (1) az időterv kritikus útját rövidítjük le, vagy (2) az erőforrások kihasználtsági fokát egyenlítjük ki.

Az időterv kritikus útvonalának lerövidítése részben ugyan időtervezési probléma, de lényegileg az erőforrás-tervezéssel is összefügg. Az alábbiakban röviden összefoglalva - többnyire csak felsorolásszerűen - azokat a megoldásokat említjük, amelyek segítségével általában eredményesen csökkenthető egy projekt időtervének kritikus útvonala.

- több és/vagy nagyobb teljesítőképességű erőforrás alkalmazása, amely a kritikus útvonal rövidítésének legegyszerűbb, de valószínűleg a legköltségesebb módja is (például pótlólagos programozói kapacitások bevonása);
- a kritikus tevékenységek felbontása több, párhuzamosan is végezhető részre, ahol ez lehetséges. Ez a megoldás minden bizonnyal több erőforrást köt le, noha rövidebb időre. (például programmodulok párhuzamos fejlesztése. Ez azonban az erős egymásra hatások miatt csak a fejlesztők közötti intenzívebb kommunikációval és a modulok elkészültét követő hangsúlyozottabb integrációs teszteléssel lehetséges);
- az időterv tevékenységei közötti átfedések számának növelése, és ahol lehetséges, az átfedések mértékének növelése is. (például a felhasználói leírás és a képzési tervek részben párhuzamos készítése a rendszermodulok fejlesztésével. Ez megtehető, hiszen elvileg mindkettő a logikai rendszerterven alapul. Más kérdés, hogy a fejlesztés befejeztével frissíteni kell e dokumentációkat.)

- a hatékonyság növelése a kritikus tevékenységek teljesítésénél. Ez többnyire akkor járható út, ha a kritikus tevékenységek között van hosszú átfutási idejű, folyamatosan teljesülő tevékenység. (Ilyen esetekben a begyakorlottság kihasználásával, illetve egy tapasztaltabb szakértő alkalmazása révén időt nyerhetünk.)

Az erőforrás-kiegyenlítés a hullámzó erőforrás-szükséglet kiegyenlítettebb, azaz egyenletesebb lekötését igyekszik megvalósítani. Az erőforrások iránti fluktuáló jellegű igény sok esetben abból fakad, hogy a projekt teljesítésének időtervében az egyes tevékenységek a legkorábbi kezdési pozícióban vannak, miközben az úgynevezett nem kritikus tevékenységeknek van valamennyi csúszási időtartamuk. Ezek a tevékenységek a legkorábbi kezdés pozíciójából egészen az úgynevezett teljes csúszási időtartamuk mértékéig anélkül elmozdíthatók, hogy ezzel kockáztatnánk a projekt teljesítési időtartamát. A csúszási időtartamokat gyakorlatilag minden projekt és többnyire minden erőforrás tekintetében eredményesen felhasználhatjuk a projekt teljesítéséhez lekötött erőforrások kiegyenlítettebbé tételére.

Az erőforrás-kiegyenlítés másik, általánosan alkalmazható módszere a bérbevétel a csúcsidőszakokban, valamint a csúcsidőszakokban végzett tevékenységek egy részének vállalkozásba adása. *(Vagy külső projektek mellett folyamatosan megvalósított saját termék fejlesztése.)*

Az erőforrás-kiegyenlítést azonban nemcsak egyetlen projekten belül célszerű végrehajtani, hanem egyfelől több projekt esetén, a szervezeten belüli összes projekt között, másfelől viszont a projektek és szervezet nem projekt jellegű (például karbantartási) tevékenységeinek erőforrás-szükségei között is. A projektek közötti erőforrás-kiegyenlítést a stratégiai prioritásokhoz való viszonyítás alapján gyakorlatilag ugyanazon megoldásokkal érhetjük el, mint az egyetlen projekten belüli kiegyenlítést. Mindehhez jól használható eszközök az egyes projektek teljesítéséhez kialakított idő-és erőforrástervek.

## 2.4 Ütemezéshez használt becslési módszerek

Ahhoz, hogy az idő- és erőforrásterveket elkészíthessük, a projektet jól körülhatárolható, kisebb feladatokra kell bontani. A lebontási struktúra elkészítésére a leggyakrabban alkalmazott eljárás a WBS (Work Breakdown Structure) illetve az RBS (Resource Breakdown Structure), melyek a projekt különböző adatainak strukturálására, csoportosítására szolgálnak. A feladat lebontásának eredményeként kapott tevékenységlista erőforrásigényének meghatározására irányuló eljárásokat kétféle ismérv szerint lehet jellemezni: azok *irányultságával* és *módszerével*.

Az irányultság lehet *fentről lefelé haladó*, amikor a globális erőforrás tervet bontjuk le súlyozott formában az egyes tevékenységekre, illetve *lentől felfelé haladó*, amikor először a tevékenységek erőforrás igényét próbáljuk meg minél pontosabban meghatározni.

Az igények meghatározásához alkalmazott módszerek két markánsan eltérő osztályba sorolhatóak: *(költség)modellek* alkalmazása és *analógiák* használata. A becslés alapját szolgáltató adatok beszerzési helye lehet [2.11]: (1) tapasztalaton alapuló becslés, (2) történeti adatok elemzése, (3) külső szakértő bevonása, (4) erőforrás felhasználási standardok felhasználása és (5) képlettel leírható konkrét összefüggések.

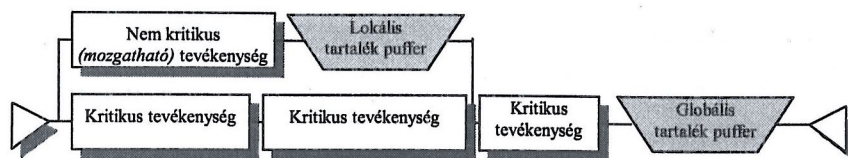
A tapasztalaton alapuló becslés nagyon hasonlít a külső szakértő bevonására, de az előbbi esetén egy belső szakértő (projektmenedzser) tapasztalataira, háttér-információira és szakértelmére alapozzuk a tevékenységek időtartamának, erőforrás szükségletének és költségének megbecslését. A külső szakértői becsléseknek (a belső viszonyok ismeretének hiányában) általában mindig át kell esniük egy adaptálási fázison. A történeti adatok felhasználása lenne az egyik legjobban megbízható eszköz, mégis kevés vállalat fordít figyelmet arra, hogy projektjeinek ilyen jellegű dokumentálását is elvégezze. Szerencsére mára már projektmenedzsmentet támogató szoftverekkel együtt számos ilyen adatbankhoz és sablonrendszerhez juthatunk hozzá (például az Oracle OLAP adatrak-



tár építési projektsablonja, vagy az SAP PM modulja). Kisebbségi projektek esetén a szabványidők és szabványosított tevékenységek alkalmazása is megoldás lehet. Ha ezeket a szabványokat nyomkövetjük, akkor ez egy alkalmas kvantitatív módszer kifejlesztéséhez vezethet. Az erőforrásigény tervezésének vannak általánosan kialakított módszerei is, melyek közül a szoftverfejlesztéshez kapcsolódókat a következő fejezetben tekintjük át részletesebben. Ezeket a módszertanokat háromféle kategóriába sorolhatjuk: (1) statikus mértéken alapuló egy vagy több változós modellek, (2) dinamikus mértéken alapuló többváltozós modellek és (3) beépített algoritmusokon alapuló elméleti modellek.

Hogy milyen előnyökkel járhat, ha a becslés folyamán magatartástudományi ismeretekre is támaszkodunk, azt egy rövid példán mutatjuk be. Gyakran tapasztalhatjuk, hogy az emberek előszeretettel építenek be biztonsági (tartalék) időket a tevékenységek hosszának megbecslésekor. Ennek az a magyarázata, hogy a projekt késedelméből származó félelmet általában nem kompenzálhatjuk az esetleges hamarabbi befejezésből származó előnyökkel. Továbbá ismert a határidő vagy más néven mérföldkő-effektus, mely szerint az emberek hatékonysága a tevékenység időtartama alatt nem állandó, és az időtartam második felének hatékonyságnövekedése az első felének hatékonyságcsökkenésével hozható összefüggésbe. Az emberi erőforrás menedzsment alkalmazásának mellőzésével könnyen elveszíthetjük a helyes mérföldkő-kijelölésből származó előnyöket.

Ha a projekt időtartamából származó kockázat az egyes tevékenységek időtartamának becslési pontosságából adódik, akkor a projekt  $\sigma$  illetve a tevékenységek  $\sigma_i$  kockázatait azok szórásaival azonosíthatjuk. Ekkor a független becslésekből származó összkockázat  $\sigma_\Sigma$  mértéke  $\sigma_\Sigma = \sigma_i \cdot \sqrt{n}$ , ami jelentősen nagyobb értéket jelent, mintha például az egész projektre nézve sikerült volna a  $\sigma = \sigma_i$  kockázati szintet biztosítani.



2.6 ábra: Lokális és globális pufferidők

A magatartástudomány ekkor is a segítségünkre lehet, hiszen a fentiek értelmében az optimista időtartambecslésnek kockázat csökkentő hatása van. Ahhoz, hogy ezt a projekttagok részéről is biztosítsuk, csak óvatosan szabad a késedelmi büntetések eszközéhez nyúlni, hiszen ezzel védekező hatást indíthatunk be. Másrészt azonban a tapasztalatok azt mutatják, hogy az emberek hajlamosak jelentősen alulbecsülni a tevékenységekhez szükséges időigényeket. Ezt tudva a projektekhez általában globálisan és nem lokálisan érdemes tartalék pufferidőt meghatározni, hiszen ez nem növeli a kockázat mértékét [2.12].

## 2.5 Erősen és gyengén korlátos (emberi) erőforrások

A sikertelen projektekből levonható következtetések közül az egyik legismertebb tévhit, hogy bármekkora mértékű késedelem korrigálható újabb erőforrások hadba állításával. Mint ahogy azt a 2.3 részben is idéztük, az emberi erőforrások pótlólagos felhasználása csak bizonyos méretek között növeli a hatékonyságot, és az általa megtakarított idő vagy behozott lemaradás átváltási rátája is változó [2.6]

Ha azt is figyelembe vesszük, hogy a megfelelő projektszemélyzet kiválasztása, a hatékony csapatépítés és csapatfejlesztés kritikus tényező abban, hogy a projekt elérje célját, akkor megállapíthatjuk, hogy egy projektcsoporthoz tagjává válni nem egyszerűen a tudás vagy a képességek kérdése. A csapatmunka támogatása azonban nemcsak a projekt kezdeti fázisában fontos, hanem végig kell, hogy kísérje az egész projekt életciklusát.

Az emberi erőforrások beszerzésének számbavételekor a lehetőségünk nem egyszerűen egy a van/nincs, alkalmas/alkalmatlan kérdés eldöntéséből áll, hi-

szen rendelkezésünkre áll például a tanulás, a tréning, vagy akár a pilot projekt eszközrendszere is. Ráadásul egy-egy ember egy adott időszak alatt több projektnek is tagja lehet.

### Definíció

*Multi-tasking* alatt értjük azt a megoldást, amikor a lokális vagy globális tartalék-idők időtartama alatt a kihasználatlan projektszemélyzetet más projekteknek kölcsönözzük.

A multi-tasking alkalmazása annak nyilvánvaló előnyei mellett rendelkezik számos hátránnyal is. Párhuzamos (multi)projektek esetében az egyik késedelme veszélyeztetheti más projektek határidőn belüli végrehajtását. A gyakori feladatkör változtatások csökkentik a hatékonyságot, és ezzel növelik az együttes költségeket. Csökkenti a csapatmunka szinergikus hatását, a szervezeti tanulás lehetőségét. Noha a hátrányok miatt a multi-tasking gyakori alkalmazása nem ajánlott, projektszervezetek esetében általában elkerülhetetlen. A kettőnél több projekt közötti multi-tasking azonban mindenképp kerülendő.

A 2.3 részben azt állítottuk, hogy az emberi erőforrások többé-kevésbé homogénnek tekinthetők. A materiális erőforrásokhoz viszonyítva ez az állítás továbbra is megállja a helyét, de a humán erőforrás igényes projektek esetén (és az informatikai projektek ilyenek) fontos a különféle jellegzetességekkel rendelkező emberi erőforráscsoportok megkülönböztetése. Ilyen osztályozások a köznyelvben is fellelhetők, amikor a szellemi (*white collar*) illetve a fizikai (*blue collar*) dolgozó megnevezéseket használjuk. A megkülönböztetés alapját eredetileg a betöltött munkakör, manapság inkább a munkavégzéshez szükséges ismeretek mennyisége és minősége határozza meg. Amikor viszont helyettesíthető vagy pótolhatatlan munkaerőt emlegetünk, akkor sokkal inkább az elvégzendő tevékenységet jellemezzük.

A szakirodalomban az *újrahasználható* illetve *nem-megújuló* erőforrás-kategóriák használata terjedt el. A megújuló erőforrások modellezése technikai szem-



pontból azért problémás, mert az ismételt felhasználhatóság mindig egyfajta rekurziót és visszacsatolást visz a rendszerbe.

Az erőforrások itt bevezetendő erősen illetve gyengén korlátos osztályozása ezeket az ismertető jegyeket ötvözi. Általában egyetlen erőforrás sem egyértelműen erősen vagy gyengén korlátos, hanem valamilyen mértékben az egyik vagy a másik tulajdonságjegyeit viseli magán. A jellemvonások közül megkülönböztetünk elsődleges jellemzőket (szürkével jelölt sorok), melyek minden erőforrás esetén értelmezettek; valamint másodlagos jellemzőket, amelyek csak emberi erőforrások esetén definiálhatóak.

2.5 táblázat: Gyengén és erősen korlátos erőforrások jellemzése

Ismertető jegy	Erősen korlátos erőforrások esetén	Gyengén korlátos erőforrások esetén
Rendelkezésre álló erőforrás egység	Gyakorlatilag véges	Elméletileg végtelen
Projekt szempontjából kulcs erőforrás	Igen	Nem
Megújítható erőforrás	Nem	Igen
Létszám- vs. időigény átváltási ráta	Alacsony	Magas
Létszámbővítés időigénye / költsége	Hosszabb / Magas	Rövidebb / Alacsony
Kieső erőforrás pótolhatósága	Nehézkés	Viszonylag könnyű
Tanulási görbe	Lapos	Meredek
Fejlődési képesség és igény	Magas	Alacsony
Végzettség, ismeretek	Magas	Alacsony
Vezetői kompetenciák	Rendelkezik	Hiányoznak
Illeszkedő vezetői stílus	Lágy	Erős
Hatáskör delegálás	Lehetséges	Kerülendő
Konfliktustűrő képesség	Alacsony	Magas
Kreativitási képesség	Magas	Alacsony
Szervezetnél eltöltött tapasztalati idő	Hosszú	Rövid
Munkahely, munkakör biztonsága	Biztos	Bizonytalan
Munkaerő piaci versenyhelyzete	Gyenge verseny	Erős verseny
Csapatmunkás jelleg	Jellemző	Nem jellemző
Multi-tasking megszakítási intervallum	Ritka	Sűrű
Életpálya orientáltság	Gazdasági	Technikai
Tervezési gondolkodás	Hosszútávú	Rövidtávú
Karrierpálya orientáltság	Párhuzamos, alternatív	Kizárólagos, versengő
Eredmény kontroll	Belső	Külső
Időorientáció	Jövő	Jelen és múlt
Feladat orientáció	Proaktív	Passzív / Reaktív
Alkalmazotti orientáció	Participatív	Paternalisztikus
Munkaerő kölcsönzési irány	Outsourcing	Insourcing
Tevékenység kreativitás igénye	Magas	Alacsony
Tevékenységhez tartozó motiváció	Pozitív	Semleges
Tevékenység monotonitási szintje	Alacsony	Magas

A 2.5 táblázatban felsorolt jellemvonások nem kizárólag csak az erőforrást, vagy csak az elvégzendő tevékenységet jellemzik, hanem sokkal inkább a kétőt egyszerre azáltal, hogy az egymásnak megfelelő képességeket és kompetenciákat írják le. Az egyes szempontok súlya eltérő, sőt projektenként is változó lehet. Ezért, hogy végül is egy adott erőforrás melyik osztályba sorolható, függ magától a projekttől is.

## INFORMATIKAI PROJEKTEK MENEDZSELÉSE

Az informatikai projekteknek van néhány olyan sajátossága, amely az általános projektmenedzsment egyes elemeinek más hangsúlyt ad. Az informatikai projektek [2.16] közül kiemelten foglalkozunk a szoftverfejlesztési projektek jellegzetességeivel.

### **Definíció**

*Informatikai projektek* alatt értünk minden olyan, az adott szervezet stratégiai céljainak megvalósítása érdekében kialakított projektet, amely az elérendő projekteredmény tekintetében informatikai megoldások bevezetését, fejlesztését illetve továbbfejlesztését valósítja meg.

### **Definíció**

*Szoftverfejlesztési projekteknek* nevezzük azokat az informatikai projekteket, amelyeknél a projekteredmény egy új vagy javított számítógépes alkalmazás.

Az informatikai projektek nem feltétlenül követelik meg számítástechnikai eszközök alkalmazását, és legtöbbször az 1. fejezetben felsorolt projekt típusok (beruházási, innovációs és szellemi szolgáltatás) egyfajta keverékei. Míg ezzel szemben a szoftverfejlesztési projektek általában olyan, többnyire belső projektek, melyek megvalósítása számítógép-használatot igényel (legalábbis az alkalmazás megépítésének végső fázisában biztosan), valamint nagyon ritkán beruházási jellegűek.

Az informatikai projektek kivitelezése többnyire eseményszerűen teljesülő tevékenységek láncain keresztül történik. Ez lényegében azt jelenti, hogy a tevékenységek készültségi fokának megállapítása problematikus. Egy programmo-



dul fejlesztése során elvégzett teljesítmény jobbára csak a tevékenység lezárása körül válik mérhetővé és láthatóvá. Ez nemcsak a projekt kontroll feladatát nehezíti, hanem jelentősen megnöveli a projekt bizonytalansági faktorát is. A bizonytalanságok egyéb forrásait és azok megnövekedett mértékét könnyen beazonosíthatjuk, ha az informatikai projekteket összevetjük a gyáripari tömegtermelés jellegzetességeivel. [2.16]

3.1 táblázat: Bizonytalanságok forrásai a projektekben

Gyáripari tömegtermelés	Informatikai rendszerfejlesztés
A termék előállítása fix telephelyen, gyakorlatilag (legalábbis rövidtávon mindenképp) változatlan körülmények között történik	A projekteredmény létrehozásának helyszíne és körülményei legtöbbször projektről projektre változnak.
A végtermék teljes mértékben standardizált.	A végtermék – a működőképes rendszer – minden esetben egyedi, jóllehet néhány eleme standardizált.
A termék előállítás és az értékesítési folyamatok időben és térben elválnak egymástól. A gyártás mindig megelőzi az értékesítést, a gyártás (néhány kivételtől eltekintve) rendszerint raktárra gyártás.	A megvalósítás és az értékesítés folyamatai gyakorlatilag időben egybeesnek, a rendszert, elkészülte után, a megrendelő azonnal használatba veszi.
Az adásvételi aktus a termék fizikai előállítása után jön létre.	Az adásvételi aktus (a szerződés) megelőzi a rendszer fizikai megvalósítását.
A termék létrehozását az eladó (a gyártó) kezdeményezi.	A rendszer létrehozását a vevő (a megrendelő) kezdeményezi.
A termékek rendszerint széles skálán helyettesíthetők egymással, s így a vevők széles határok között választhatnak az ár, a minőség, a funkciók stb. tekintetében. A választás elsődleges kritériuma rendszerint az ár.	A helyettesíthetőség, és így a vevő választási lehetőségei is erősen korlátozottak. A választásnak csak egyik, de nem feltétlenül elsődleges kritériuma az ár.

### 3.1 Szoftver- és rendszerfejlesztési projektek

Ha a 3.1 ábrát tekintjük, akkor láthatjuk, hogy minden szoftverfejlesztéssel foglalkozó cég elsődleges stratégiai jellemzője a tervezési, termékfejlesztési, egyedi alkalmazásfejlesztési és kutatási profilokból alkotható portfólión alapul. Az egyes profilok elemei által meghatározott szoftverfejlesztési projektek szerkezete kevésbé tér el egymástól. [3.1]. Az így kialakított üzleti stratégiának közvetlen hatása van az informatikai stratégia három fő elemére: (1) az adat- és alkalmazási stratégiára, (2) a technológiai stratégiára, valamint (3) az informatikai szervezeti stratégiára.



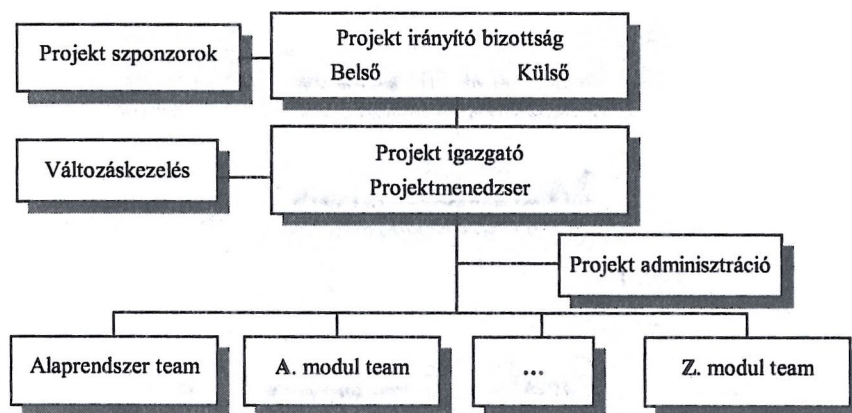


### 3.2 A szoftverfejlesztési projektek speciális aspektusai

A szoftverfejlesztési projekteknel minden fázisban nagyon fontos a tervezés, de az egész folyamatot átfogó iteratív tervezés is nélkülözhetetlen, ugyanis az előforduló hibák javításának költsége nagyságrendekkel olcsóbb a korai fázisokban, mint ha ugyanaz egy későbbi fázisban merül fel.

Ideális esetben a fejlesztési követelményeket a projektmenedzserek határozzák meg, a felhasználói és technikai specifikációk ismeretében. A szoftverfejlesztéssel foglalkozó projektszervezetek felépítése is jellegzetes, melynek általános vázát a 3.3 ábra organigramja mutatja [3.5].

A szoftverfejlesztési projektek sikerességének egyik kulcskérdése az, hogy az éppen aktuális projekt részeredményei milyen mértékben hasznosíthatóak újra, illetve mennyire épülnek be a következő projektekbe. Mivel általában nincs két egyforma szoftverfejlesztési projekt, ezért maga a projekt egy egyéni és szervezeti tanulási folyamatként is működtethető [3.3].



3.3 ábra: Szoftverfejlesztési projektek általános szervezeti felépítése

Szoftverfejlesztés esetén a szükséges technológiai és technikai háttér kialakítása általában nem okoz problémát, éppen ezért fokozottabb szerepet kap az emberi erőforrás oldal. A projektmenedzserek vezetői képességei kritikus sikertényezőkké válnak. A siker okaként leggyakrabban az alábbi vezetői tulajdonságok emelhetők ki: szervezeti és szellemi katalizátori képességek, kariz-



matikus egyéniség, teljesítmény megítélő képesség, kivételes helyzetek gyors kezelése, aktív felügyelet és támogatás.[3.6]

A szoftverfejlesztésben résztvevő személyzetnek egy speciális, egyirányú helyettesíthetőségi rendszere van. A projekt életciklusában előbb ütemezett és erősebben korlátos személyzet helyettesíteni tudja a később és gyengén korlátos erőforrásokat. Például egy programozó szükség esetén tesztelési feladatokat is elláthat. Ez egyben azt is jelenti, hogy pótlólagos erőforrások bevonása nemcsak kívülről lehetséges, hanem belső funkciók szerinti „*visszaléptetéssel*” is.

A szoftverfejlesztési projektek legjellegzetesebb vonása mégis az, hogy nagyon nehéz megbecsülni a kivitelezéshez szükséges erőforrás-mennyiséget. A köztudatba „*szoftverkrízisként*” bevonult jelenség egyik kiváltó oka éppen ez a tény volt. Noha a módszerek mára sokat fejlődtek, még manapság sem ritka, hogy a becslések akár 200%-kal is eltérnek a végső, valós adatoktól, és a sikertelen projektek aránya is 30%-ra becsülhető [3.2]. A helyzet javítására számos erőforrás-tervezési modell készült, melyek közül - kizárólag az illusztráció kedvéért - a két legrégebbit nagyvonalakban ismertetjük is.

A „*kódolási idő háromszorosa*” elnevezés egy nagyon egyszerű eljárást takar, amit előszeretettel alkalmaznak még ma is, főleg gyorsasága és olcsósága miatt. Az ökölszabály a következő: állapítsuk meg a kódolás várható idejét, majd szorozzuk meg hárommal, és megkapjuk azt a becsült időt, amely alatt a tervezés, a tesztelés és a dokumentálás is elvégezhető.

A *COCOMO* modellt eredetileg nagyméretű projektekhez dolgozták ki, de jól alkalmazható kisméretű projektek esetén is. A projekt feladatait olyan szintre kell lebontani, amelyekre a szükséges erőforrásigény az előző tapasztalatok alapján már megbecsülhető. Ezen szegmensek mérése a KDSI egységekben történik, amely egy egységének ezer forráskódú utasításszám felel meg. A módszer 15 költségparaméter és két modellparaméter segítségével határozza meg a projekt kivitelezésének időigényét.

### 3.3 Egyszerűsítések kisebb szoftverfejlesztési projektek esetén

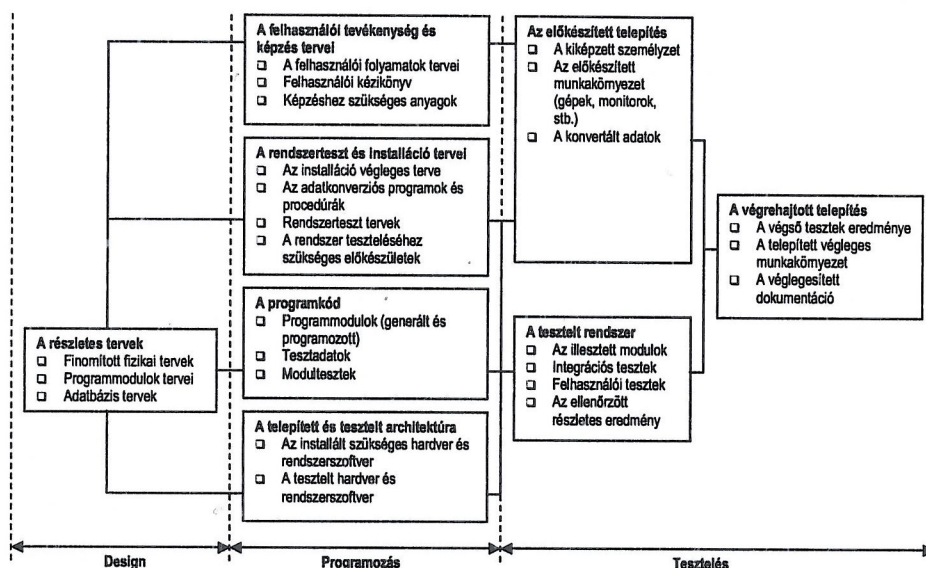
A 4-7. fejezetekben ismertetendő modell méretének kezelhetősége érdekében néhány megszorításhoz és egyszerűsítéshez kell folyamodnunk. Ha a rendszerfejlesztés folyamatát a spirális életciklus modell egyes körívei mentén szétbontjuk, akkor mindig egy „specifikáció – tervezés – kivitelezés – ellenőrzés” lépés sorozatot kapunk. Ha feltételezzük, hogy kezdetben a specifikációk eleve rendelkezésre állnak, akkor a 3.4 ábrán feltüntetett általános rendszerfejlesztési struktúra [2.16] vízszintes időtengelye mentén ugyanilyen lépéseket azonosíthatunk. A felbontás során így kapott lépéseket megbonthatatlan egységként kezeljük, melynek következtében több, párhuzamosan végrehajtható alprojektet képezhetünk.

A rendelkezésre álló erőforrásokat ennek megfelelően három típusba soroljuk. A programtervezők (designerek) képviselik a tervezési fázishoz rendelhető erőforrásokat. A tervezési munkafázis elvégzéséhez magas szakmai tapasztalat és szervezési ismeretek szükségesek. A 2.3 táblázatból egyértelműen leolvasható, hogy ez erősen korlátos erőforrástípusként kezelendő.

Az alprojektek utolsó fázisát a tesztelők végzik. Sok alkalmazás esetében a tesztelés a programozáshoz hasonló, komoly kihívást jelent a személyzet számára, zömében azért mégis egy adott forgatókönyv szerint kell a terméket bevizsgálni. A hibakeresés és ellenőrzés monoton munka, nem igényel rendszerismeretet, valamint semleges, sőt kritikus attitűdöt feltételez, ezért gyenge erőforrásként kezelhető.

Az egyes rendszerterv elemek kódolását a programozók végzik. A programozási feladatok, illetve az elvárt képességek ezt az erőforrásfajtát valahova éppen a gyenge és erős erőforrások közé pozícionálják. Amennyiben azonban a tervezés minősége magas színvonalú, a kódolás folyamata „mindössze” egy adott programnyelv és a hozzátartozó fejlesztői környezet ismeretét feltételezi. Gyenge erőforrásként történő kezelése ennek ellenére önkényesnek tekinthe-

tő, és mindössze a több kritérium szerinti optimalizálás bemutathatóságát célozza [3.4].



3.4 ábra: Egy alkalmazói rendszer általános funkcióhordozó struktúrája

Az alkalmazott esettanulmányok során mindig feltételeztük, hogy a technikai specifikáció elkészítése a tervezési fázisban történik, a dokumentációkészítés pedig a tesztelési fázis része. A tevékenységeket és azok erőforrásigényét pontos adatokként kezeljük, bármilyen eljárással és megbízhatósági szinten lettek is előállítva.

Mindezen megkötések kisméretű (*small-scale business application*) fejlesztési feladatok esetén [3.7] megállják a helyüket, és a gyakorlatban is alkalmazható modell kialakítását teszik lehetővé.



## TEVÉKENYSÉGI HÁLÓK, HÁLÓTERVEZÉS

Az ütemezési problémák gyökerüktől fogva magukban hordozzák a bonyolultságot. Ennek megfelelően az összes itt tárgyalt feladat az NP nehézségű problémaosztályba tartozik. Régebben ezért főleg a vizuális tervezés eszközeit alkalmazták. A számítógépek megjelenésével mód nyílt az ütemezéstámogatás gépesítésére is. Ezt a nehéz problémát a mai szoftverek (*MS Project*, *CA Visual Project*, *Prochain*, *Concerto*, *Project Scheduler*) úgy oldják meg, hogy mindig a legkorábbi (vagy legkésőbbi) kezdési időpontokból kiindulva heurisztikus módszerekkel próbálkoznak egy helyes ütemezést előállítani. A felsorolt programok fejlődése is jól jellemzi, mekkora a piaci igény arra, hogy projekt-ütemezésekből származó feladatok megoldását támogassuk, egyszerűsítsük. Ne feledkezzünk meg azonban arról, hogy ezek nem mindenható eszközök, és ráadásul a hatékony ütemezés előállítása a projektek sikerességének technikai kulcstényezői közül a mindössze csak az egyik.

## 4.1 Időmenedzsment és ütemezés-tervezés

A projekt időtervének vagy ütemezésének elkészítése az a feladat, amely során a projekteredmény eléréséhez szükséges tevékenységek időbeli kezdetét, végét és összefüggéseit meghatározva olyan program áll elő, amely útmutatást ad a projekt határidőn belüli sikeres végrehajtásához. Az ütemtervek részletezettségi igénye az egyes döntéshozói szinteken eltérő lehet, és általában mindig grafikus ábrázolásmód segítségével történik. Az ütemterv nemcsak a projekt kivitelezési programjának leírásaként funkcionál, hanem egyfajta kommunikációs mód is. A különféle erőforrás felhasználási sáv és oszlopdiagramok, a precedencia diagram, az időskálázott CPM diagram vagy a leggyakrabban

használt függőségi nyilakkal kiegészített Gantt diagram is az ütemezési információk egyes csoportjait úgy emeli ki, hogy közben másokat elhanyagol, vagy rosszabb esetben eltorzít. Ennek ellenére használatuk sokszor nélkülözhetetlen ahhoz, hogy a projektet könnyen áttekinthessük.

Az ütemterv készítésének egyik lehetséges módját a PMBOOK módszertan [1.1] C oszlopának eggyel mélyebb szintű kibontásával tekinthetjük át.

4.1 táblázat: A projektmenedzsment időgazdálkodáshoz kapcsolódó funkciói

Időmenedzsmenthez kapcsolódó funkciók projektekben		
(C1) Tevékenység definiálás	(C2) Függőség meghatározás	(C3) Időigény becslés
<p>A projekt végrehajtásához szükséges tevékenységek azonosítása és meghatározása.</p> <p><b>Inputok:</b></p> <p>(C1-I1) <i>Lebontási struktúra</i> - WBS fa struktúrája</p> <p>(C1-I2) <i>Kiterjedés tervek</i> - menedzselhető célok listája</p> <p>(C1-I3) <i>Történeti adatok</i> - előző projektek jegyzőkönyvei</p> <p>(C1-I4) <i>Korlátok listája</i> - megszorítások, melyek behatárolják a projekt megvalósítását</p> <p>(C1-I5) <i>Feltételek listája</i> - tervezéskor figyelembe veendő valós vagy vélt tényezők (kockázatok)</p> <p><b>Eszközök és módszerek:</b></p> <p>(C1-E1) <i>Dekompozíció</i> - a projekt elemeinek kisebb, menedzselhető komponensekre bontása</p> <p>(F1-E2) <i>Sablonok</i> - más projektek újra felhasználható elemeinek alkalmazása</p> <p><b>Outputok:</b></p> <p>(C1-O1) <i>Tevékenységi lista</i> - az atomi tevékenységek teljes felsorolása</p> <p>(C1-O2) <i>Támogató segédinformációk</i> - a projekthez azonosított korlátok és feltételezések dokumentációja</p> <p>(C1-O3) <i>Finomított WBS</i> - a javított projekt lebontási struktúra</p>	<p>A tevékenységek közötti közvetlen és közvetett függőségek meghatározása.</p> <p><b>Inputok:</b></p> <p>(C2-I1) <i>Tevékenységi lista</i> - lásd (C1-O1)</p> <p>(C2-I2) <i>Termék specifikáció</i> - termék karakterisztikai technológiai függőségeket jelenthetnek</p> <p>(C2-I3) <i>Természetes függőségek</i> - alkalmazott technológiákban rejlő folyamatfüggőségek</p> <p>(C2-I4) <i>Mesterséges függőségek</i> - a projektvezetés által definiált függőségek (pl. best practices)</p> <p>(C2-I5) <i>Külső függőségek</i> - függőség projekten kívüli tényezőktől</p> <p>(C2-I6) <i>Korlátok listája</i> - lásd (C1-I4)</p> <p>(C2-I7) <i>Feltételek listája</i> - lásd (C1-I5)</p> <p><b>Eszközök és módszerek:</b></p> <p>(C2-E1) <i>Precedencia diagramok</i></p> <p>(C2-E2) <i>Logikai diagramok</i></p> <p>(C2-E3) <i>Hálótervezés diagramok</i></p> <p>(C2-E4) <i>Hálótervezési sablonok alkalmazása</i> - lásd 4. fejezet, 4.2 szakasz</p> <p><b>Outputok:</b></p> <p>(C2-O1) <i>Projekt hálóterve</i> - a tevékenységek és a közöttük lévő függőségek grafikai ábrázolása</p> <p>(C2-O2) <i>Finomított tevékenységi lista</i></p>	<p>A tevékenységek végrehajtásához szükséges, valószínűsíthető időtartamok megállapítása.</p> <p><b>Inputok:</b></p> <p>(C3-I1) <i>Tevékenységi lista</i> - lásd (C1-O1)</p> <p>(C3-I2) <i>Korlátok listája</i> - lásd (C1-I4)</p> <p>(C3-I3) <i>Feltételek listája</i> - lásd (C1-I5)</p> <p>(C3-I4) <i>Erőforrásigény lista</i> - a tevékenységek human és materiális erőforrásainak szükséglete</p> <p>(C3-I5) <i>Erőforrások szakmai leltára</i> - lásd 2.3 táblázat</p> <p>(C3-I6) <i>Történeti adatok</i> - projekt jegyzőkönyvek, külső szakmai adatbázisok, szervezeti tudás dokumentumai</p> <p><b>Eszközök és módszerek:</b></p> <p>(C3-E1) <i>Szakértői becslések</i></p> <p>(C3-E2) <i>Analógiák alkalmazása</i></p> <p>(C3-E2) <i>Szimulációk</i> - lásd 2. fejezet, 2.3 szakasz</p> <p><b>Outputok:</b></p> <p>(C3-O1) <i>Időigény lista</i> - az atomi tevékenységek várható időigénye (és becslési pontossága)</p> <p>(C3-O2) <i>Becslés folyamatának dokumentációja</i> - a becslés során felhasznált feltételrendszer rögzítése</p> <p>(C3-O3) <i>Finomított tevékenységi lista</i></p>

(C4) Ütemezés	(C5) Ütemezés felügyelet
<p>A tevékenységek megkezdésének és befejezésének meghatározása.</p>	<p>Az ütemezés végrehajtásának biztosítása és nyomon követése, a változások menedzselése.</p>
<p><b>Inputok:</b>  (C4-I1) <i>Projekt hálótérve</i>  - lásd (C2-O1)  (C4-I2) <i>Időigény lista</i>  - lásd (C3-O1)  (C4-I3) <i>Erőforrásigény lista</i>  - lásd (C3-I4)  (C4-I4) <i>Erőforrások rendelkezésre állási profiljai</i>  - melyik erőforrás, mikor és milyen mennyiségben érhető el a projekt időtartama alatt  (C4-I5) <i>Erőforrás naptárak</i>  - projektet befolyásoló időpontok jegyzéke (ünnepek, tanfolyamok stb.)  (C4-I6) <i>Mérföldkövek jegyzéke</i>  (C4-I7) <i>Feltételek listája</i>  - lásd (C1-I5)  (C4-I8) <i>Pufferidők listája</i>  - kötelező leállások, szerviz és pihenőidők</p>	<p><b>Inputok:</b>  (C5-I1) <i>Ütemterv</i>  - lásd (C4-O1)  (C5-I2) <i>Teljesítménymérési jegyzőkönyvek</i>  - a projekt részarányos teljesítésének dokumentumai  (C5-I3) <i>Változtatási kérelmek</i>  - az ütemtervet is érintő változások  (C5-I4) <i>Ütemezés változási terv</i>  - lásd (C4-O3)</p>
<p><b>Eszközök és módszerek:</b>  (C4-E1) <i>Matematikai analízis</i>  - CPM, GERT, PERT módszerek  (C4-E2) <i>Időtartam rövidítés</i>  - párhuzamosítás vagy pótlólagos erőforrások bevonása  (C4-E3) <i>Szimuláció</i>  (C4-E4) <i>Erőforrás kiegyenlítési algoritmusok alkalmazása</i>  (C4-E5) <i>Projektmenedzsment szoftverek használata</i></p>	<p><b>Eszközök és módszerek:</b>  (C5-E1) <i>Ütemezés kontroll</i>  - a változtatás engedélyeztetésének szabályzati rendszere  (C5-E2) <i>Teljesítménymérés</i>  - a projekt részarányos teljesítésének megállapítása  (C5-E3) <i>Újratervezési módszerek</i>  - erőforrás igények pontosítása, alternatív megoldások integrálása  (C5-E4) <i>Projektmenedzsment szoftverek használata</i></p>
<p><b>Outputok:</b>  (C4-O1) <i>Ütemterv</i>  - vizualizálható és könnyen frissíthető formában  (C4-O2) <i>Támogató segédinformációk</i>  - erőforrás-kihasználtsági hisztogramok, alternatív ütemezések  (C4-O3) <i>Ütemezés változási terv</i>  - az ütemtervtől való eltérések korrekciójának szabályozása  (C4-O4) <i>Javított erőforrásigény lista</i>  - az erőforrás-kiegyenlítés eredményeinek visszavezetése</p>	<p><b>Outputok:</b>  (C5-O1) <i>Frissített ütemterv</i>  - a kérelmeknek és igényeknek megfelelő változtatások visszavezetése  (C5-O2) <i>Korrekciós intézkedések</i>  - a projekt eredeti ütemtervéhez való visszatérés érdekében tett lépések listája  (C5-O3) <i>Tapasztalatok rögzítése</i>  - a projekt ütemezésével, illetve a korrekciós intézkedések hatásával kapcsolatos tanulságok dokumentálása</p>

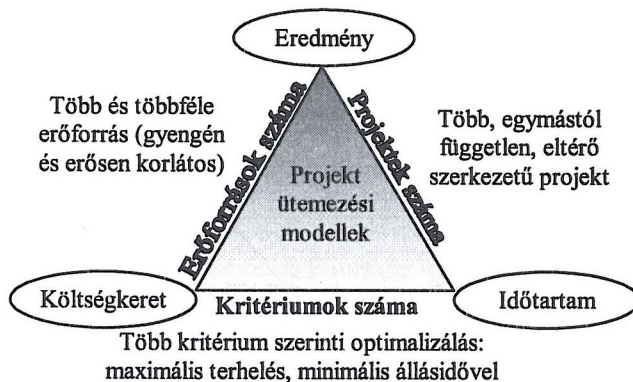


A továbbiakban kizárólag az ütemterv-készítés (C4) eszközrendszerével, azon belül is a (C4-E4) erőforrás hozzárendelési és kiegyenlítési algoritmusok problémakörével foglalkozunk.

## 4.2 Erőforrás hozzárendelési és kiegyenlítési problémák

A determinisztikus tevékenységi hálók esetén feltételezzük, hogy a feladat összes paramétere pontosan ismert. Az ilyen hálók tervezési problémáit a 4.1 ábrán látható három dimenzió alapján lehet kategorizálni. Az ábrában előre feltüntetettük az ismertetendő, erős és gyenge erőforrásokat kezelő modellünk jellegzetességeit is. [4.2]

Ha a tevékenységek megfelelő időrendű ütemezése mellett egyéb, a tevékenységekhez köthető erőforrások felhasználását is menedzselni kell, akkor erőforrás hozzárendelési feladatról beszélünk. Amennyiben az erőforrás felhasználási profil alakjával kapcsolatban előzetes elvárásaink vannak, akkor erőforrás kiegyenlítési feladattal állunk szemben.



4.1 ábra: A determinisztikus ütemezési problémák kategóriarendszere

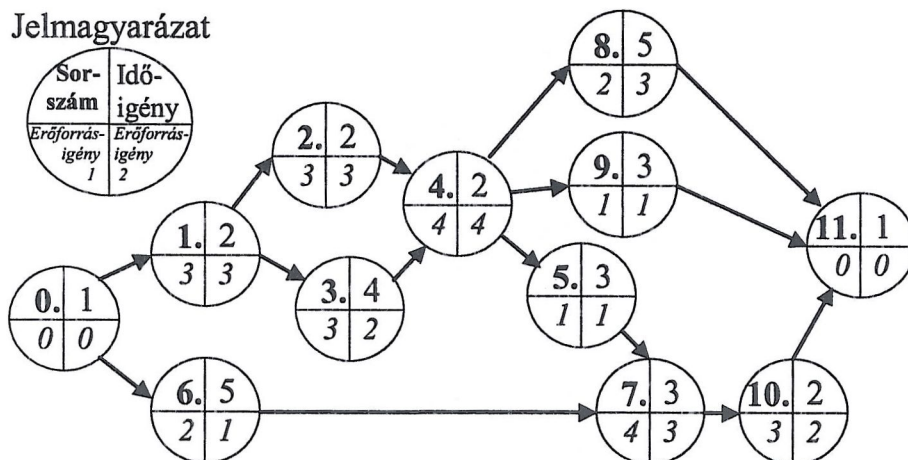
A többprojekttes problémák visszavezethetők az egyprojekttes esetekre, ha az összes projektet megelőzően, illetve követően bevezetünk egy üres (idő- és erőforrásigény nélküli) tevékenységet, és az összes projektet a kettő közé illesztve egy „szuperprojektet” képezünk. A többprojekttes problémák lehetnek homogének illetve inhomogének aszerint, hogy a benne szereplő projektek

ugyanazon struktúrájú projektekből állnak vagy sem. Homogén projektek esetében minden projekt lényegében egyazon projekt időben eltoltságaiból áll, míg az inhomogén esetben a projektek igen eltérőek lehetnek. A szoftverfejlesztési feladatok változatossága miatt nem használhatjuk ki a homogén struktúrák adta speciális struktúrából fakadó előnyöket.

A következő fejezetekben először mindig egy - a 4.1 ábrán feltüntetett kategóriák szerinti - determinisztikus, egy kritériumos, egy projekt, de több erőforrást kezelő alapmodellt ismertetünk, amelyen keresztül bevezetünk egy újfajta „globális” kritériumot: az állásidő mértéket. Az alapmodell ismertetése után kerül sor a gyengén és erősen korlátos erőforrások kezelésére is alkalmas, több projekt és több kritériumos kiterjesztett modellek bemutatására.

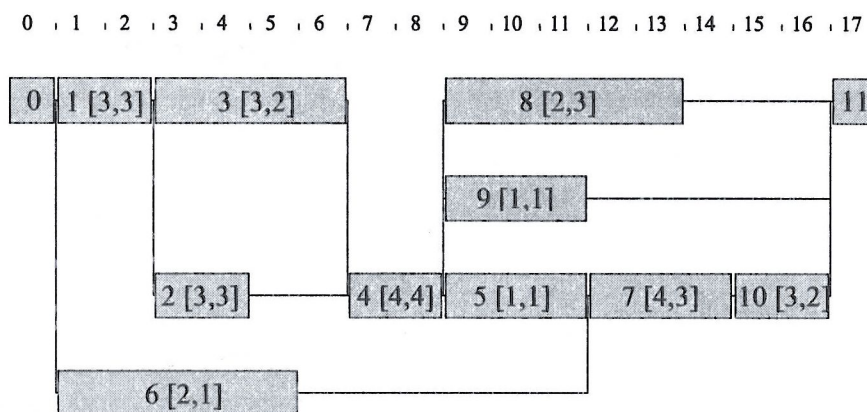
Az erőforrás allokációs modellek szempontjából a projekt egymástól függő tevékenységek halmaza, ahol minden tevékenységhez adott annak idő- és erőforrásigénye. Az ilyen jellegű feladatok ábrázolására a legelterjedtebb módszer az AoN (Activity on Node) gráfok használata, melyekben az egyes tevékenységeket a gráf csúcspontjaival, a közöttük lévő precedenciákat pedig irányított élekkel jelöljük.

#### Jelmagyarázat



4.2 ábra: Egy ütemezési feladat (ESS) leírása AoN diagrammal

A feladatok leírására itt az AoN gráfok egy speciális változatát, a kiterjesztett Gantt diagramot használjuk. A kiterjesztett Gantt diagramokon az egyes csúcspontokat téglalapokkal helyettesítjük, ahol a téglalapok hossza jelenti az adott tevékenység időigényét. A téglalapok elhelyezése a vízszintes időtengely mentén, rendezetten történik oly módon, hogy a téglalapok bal szélé a kezdési időpontot, a jobb szélé pedig a befejezési időpontot jelöli ki. A téglalapba írva a tevékenységek sorszáma után zárójelben azok erőforrásigényét, míg a közöttük fennálló közvetlen megelőzési relációkat összekötő vonalakkal adjuk meg. [4.1]



4.3 ábra: Az ESS ütemezési feladat leírása kiterjesztett Gantt diagrammal

Rekurziómentes projektek esetén a következő lemma mindig alkalmazható, lévén az AoN gráfjaik körmentesek. Rekurziót tartalmazó projekteknel a rekurzív tevékenységsorozatokat mindig helyettesíthetjük azok bizonyos mélységű lineáris kifejtésével. Rövidtávú projekteknel ez nem jelent megszorítást.

#### 4.1 Lemma

Egy véges, körmentes  $G = (V, E)$  (aciklikus) irányított gráf csúcsai mindig beszámozhatóak az  $1 \dots |V|$  számokkal úgy, hogy minden  $(i, j) \in E$  él esetén teljesül az  $i < j$  egyenlőtlenség.



**Bizonyítás:** A bizonyítást teljes indukcióval végezzük. Az állítás az összes egy csúcspontú gráfra triviálisan teljesül. Tegyük fel, hogy az állítás minden  $n$  csúcsú körmentes gráfra is teljesül. Ekkor tekintsünk egy tetszőleges  $n+1$  csúcspontú körmentes gráfot. A csúcsok között biztosan van olyan, amelybe csak bevezető élek vannak. Ellenkező esetben minden csúcsból vezetne ki él, vagyis tetszőleges csúcsból elindulva mindig tudnánk folytatni az utunkat, ami legfeljebb  $|V|$  lépésben körhöz vezetne. Ezt a csúcspontot a bevezető élekkel együtt kitörölve egy  $n$  csúcsú, körmentes gráfot kapunk, amelynek csúcsai az indukciós hipotézis szerint már megfelelően beszámozhatóak. Ha a törölt csúcsponthoz az  $n+1$  értéket rendeljük, akkor egy jó számozást kapunk. ♦

A 4.1 lemmának megfelelően a továbbiakban mindig feltételezzük azt, hogy ha két tevékenységet irányítatlan él köt össze, akkor a kisebb sorszámúnak kell megelőznie a nagyobb sorszámút.

### 4.3 Jelölések

Jelölje  $P = \{P_1, P_2, \dots, P_n\}$  az adott időszak alatt elvégzendő projektek halmazát. A rendelkezésünkre álló  $k$ -féle erőforrásból álló halmazt jelölje  $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ . Ha az  $R \in \mathcal{R}$  erőforrás korlátos, akkor adott az  $R(t)$  függvény, amely minden időpontban megadja az  $R$  erőforrásból rendelkezésre álló mennyiséget. Minden  $P_i = \{T_{i,j} : j = 1 \dots n_i\}$  projekt  $n_i$ -féle tevékenységet tartalmaz.

Az eddig bevezetett jelölések segítségével az  $i$ . projekt  $j$ . tevékenysége egy  $(D_{i,j}, R_{i,j}^1, R_{i,j}^2, \dots, R_{i,j}^k)$  vektorral jellemezhető, ahol  $D_{i,j} \in \{1, 2, \dots\}$  az adott tevékenység időigényét, míg  $R_{i,j}^r \in \{1, 2, \dots\}$  pedig az adott tevékenységnek az  $r$ . erőforrásra vonatkozó erőforrásigényét jelöli.

Az AoN gráf éleinek megfelelő  $(T_{i,j}, T_{i,l}) \in E$  közvetlen megelőzési relációk azt jelentik, hogy az  $i$ . projekt  $j$ . tevékenységének még az  $i$ . projekt  $l$ . tevékenységének megkezdése előtt be kell fejeződnie. Ez a jelölés egyenértékű az  $S_{i,j} + D_{i,j} \leq S_{i,l}$  egyenlőtlenség teljesülésével, melyre a továbbiakban az  $(i, j) \rightarrow (i, l)$  rövidítést használjuk.

Ha a  $T_{i,j}$  tevékenység ütemezésének kezdetét  $S_{i,j} \in \{0, 1, \dots\}$ -vel jelöljük, akkor az összes projekt egy  $S$  ütemezését a tevékenységek kezdési időpontjainak vektorával adhatjuk meg  $S = (S_{i,j} : i = 1 \dots n, j = 1 \dots n_i)$ . Gyakran nincs szükségünk az egész vektor megadására (például ha csak a nem kritikus tevékenységek ütemezését akarjuk megadni). Ilyenkor az alkalmazott szoftverek jelöléseinek megfelelően az  $S_{i,j} = t$  írásmód helyett az egysoros  $(i, j) \Rightarrow t$  rövidítést használjuk annak leírására, hogy az  $i$ . projekt  $j$ . tevékenysége a  $t$ . időpontban kezdődik.

Az összesen  $N = \sum_{i=1}^n n_i$  tevékenységet tartalmazó több projekt esete mindig visszavezethető az egy projekt esetre, az egyes projektek párhuzamos sorbakapcsolásával. Bár ez a megoldás nem minden esetben a leghatékonyabb - hiszen az egyesített szuperprojekt mérete jelentősen nagyobb, mint az egyes alprojektek mérete -, a jelölésrendszert mégis jelentősen leegyszerűsíti. A több projekt párhuzamos összekapcsolása úgy történik, hogy bevezetünk egy 0. nyitó, valamint egy  $N+1$ . záró, erőforrásokat nem használó, üres tevékenységet, ahol  $D_0 = D_{N+1} = 1$  és  $R_0^j = R_{N+1}^j = 0, j = 1 \dots k$ . Jelölje  $\bar{P}_i = P_i - \{T_{i,j} : \exists l (i, l) \rightarrow (i, j)\}$  az  $i$ . projekt azon tevékenységeinek halmazát, melyekbe nem vezet él; és hasonlóan  $\bar{P}_i = P_i - \{T_{i,j} : \exists l (i, j) \rightarrow (i, l)\}$  pedig azokat, melyekből nem vezet ki él. Ekkor minden  $\bar{P}_i$ -beli tevékenység elé rendeljük a 0. nyitó tevékenységet, és minden  $\bar{P}_i$ -beli tevékenység után pedig az  $N+1$ . záró tevékenységet. Így az üres nyitó tevékenységekhez tartozó

közvetlen megelőzési relációk halmaza  $K_0 = \{0 \rightarrow T : T \in \bar{P}_i, i = 1 \dots n\}$  lesz, valamint a záró tevékenység esetén  $K_{n+1} = \{T \rightarrow N+1 : T \in \bar{P}_i, i = 1 \dots n\}$ . Az  $i$ . projekthez tartozó közvetlen megelőzési reláció halmazát  $K_i$ -vel jelölve a szuperprojekt közvetlen megelőzési relációhalmazát így adhatjuk meg:

$$K = \bigcup_{i=0}^{n+1} K_i .$$

A két üres tevékenységet az ábrákon a „>” és a „<” jelekkel jelöltük, és az időigényüket mindössze az ábrázolhatóság kedvéért választottuk egységnyinek. Ekkor azonban ekvivalencia megtartása érdekében szükséges, hogy az egyesített projekt kezdőpontját a 0. időpontba helyezzük át. Az egy projektes alakra történő visszavezetése után a 4.1 lemmát alkalmazva a tevékenységeket egyszerűbben, azok sorszámaival ( $1 \dots N$ ) is azonosíthatjuk. Ennek megfelelően a tevékenységek jelölésére az  $(i, j)$  index helyett azok sorszámát használjuk. Jelöléseink tovább egyszerűsödnek, ha a dupla index helyett is mindig az egyszerűs, sorszám szerinti indexelést alkalmazzuk. A továbbiakban mindig ezt a konvenciót követjük.

Nem jelent tehát megszorítást, ha feltételezzük, hogy az ütemezések mindig az  $S_0 = 0$  időpontban kezdődnek. Ezen feltételezés mellett az  $S$  ütemezés végrehajtásához szükséges idő  $E(S) = S_{N+1}$ .

A számos ütemezési lehetőség közül csak a lehetséges ütemezéseket vesszük figyelembe, vagyis azokat, melyek megtartják a közvetlen megelőzési relációkat.

### Definíció

$S$  egy *lehetséges ütemezés*, ha minden  $i \rightarrow j$  közvetlen megelőzési reláció esetén teljesül az  $S_i + D_i \leq S_j$  egyenlőtlenség. A lehetséges ütemezések halmazának jele  $\mathcal{S}$ .



Egy kritérium esetén hatékony ütemezés alatt az adott célfüggvény szerinti optimumot, több kritérium esetén pedig a (nem dominálható) Pareto optimumokat értjük.

#### 4.4 A kritikus út módszere

A kritikus út módszerének (továbbiakban CPM – Critical Path Method) rövid ismertetését a történeti jelentőségén túl az is szükségessé teszi, hogy az ismertető módszerek fel is használják. A módszer helyét és lényegét a Matematikai kislexikon [4.4] az oprációkutatás címszó alatt az alábbiakban jelöli ki:

„... Az operációkutatás körébe sorolt problémák egyik közös jellemzője, hogy valamilyen adott szempontból legkedvezőbb – optimális – döntések meghozatalára vonatkoznak. – Bár az operációkutatásban alkalmazott matematikai módszerek feladatról-feladatra változhatnak, vannak bizonyos leggyakrabban használt módszerei is. Ilyen például a matematikai programozás; ez elsősorban optimális termelési, értékesítési és intézkedési tervek (programok) megalkotását teszi lehetővé. – Az operációkutatás további tipikus módszerei közé tartozik a nagyméretű intézkedési ütemtervek készítése (pl. nagyberuházások megvalósításával kapcsolatban), amiben elsősorban gráfelméleti eredményekre támaszkodnak. A legismertebb ilyen módszer lényege, hogy az egymáshoz kapcsolódó, különböző időtartamú tevékenységek közül kijelölik azokat, amelyek időtartamának összege meghatározza a teljes munkálat elvégzéséhez legalább szükséges időtartamot. E kijelölt *kritikus* tevékenységek az összes tevékenység kapcsolódását szimbolizáló gráfban egy útnak felel meg, innen adódik a kritikus út módszere elnevezés.”

A CPM eljárás az egyes tevékenységek legkorábbi ( $ES_i$ ) és legkésőbbi ( $LS_i$ ) kezdési időpontjainak meghatározásával ad választ a kritikus út kijelölésére. Ha feltételezzük, hogy az AoN diagram csúcsai a jelölésekben leírt konvenciók alapján 0-tól kezdődően  $N + 1$ -ig vannak sorszámozva, akkor ezek az értékek könnyen kiszámíthatóak az alábbi rekurzív képletek segítségével:

$$ES_0 = 0 \text{ és növekvő index szerinti rendben } ES_i = \max_{i \rightarrow j} (D_j + ES_j)$$

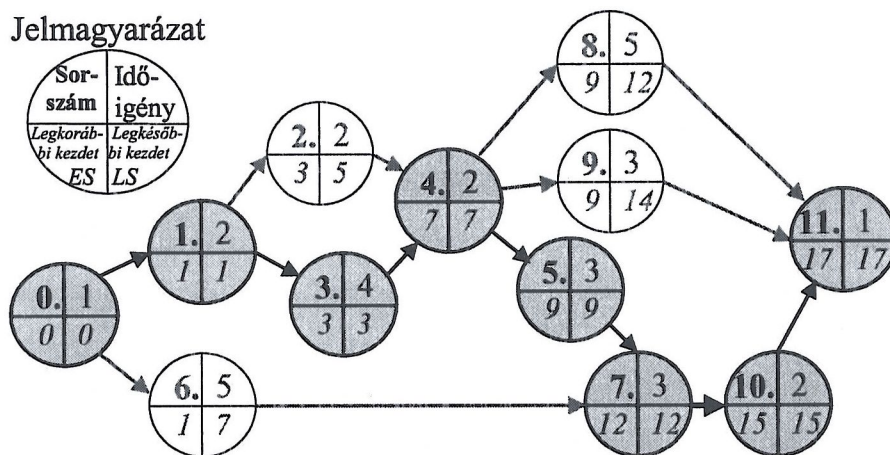
Majd ezután a számítást „visszafele” is elvégezve:

$$LS_{N+1} = ES_{N+1} \text{ és csökkenő index szerinti sorban } LS_i = \min_{i \rightarrow j} (LS_j - D_i).$$

Azokat a tevékenységeket, amelyeknél  $ES_i = LS_i$ , kritikus tevékenységeknek nevezzük. A kritikus út pedig az eredeti gráf azon éleiből rajzolódik ki, melyek két kritikus csúcspontot kötnek össze.

A közvetlen megelőzési relációk segítségével tehát minden tevékenységhez meghatározható az  $[ES_i, LS_i]$  (legkorábbi – progresszív időterv illetve legkésőbbi – retrográd időterv szerinti) tevékenységkezdeti idők intervalluma, amely az adott tevékenység lehetséges kezdési időpontjait tartalmazza [4.3].

#### Jelmagyarázat



4.4 ábra: Az ESS ütemezési feladat CPM szerinti kritikus útvonala

#### 4.5 Határidők és tartalékidők

A 2.4 fejezet kockázatbecsléssel foglalkozó részében már érintettük a lokális és globális pufferidők problematikáját. A *tartalékidő* fogalma a lokális pufferidőhöz kapcsolódik oly módon, hogy technológiai okokból néha bizonyos tevékenységek befejezése után, illetve azok megkezdése előtt kötelező várakozási időt kell betartani (például kellő kiszáradás eléréséhez). A tartalék-

idők kezelése minden további nélkül megoldható, ha olyan virtuális tevékenységeket vezetünk be, melyeknek csak időigénye van, és nem kötődik hozzájuk erőforrás felhasználás. Ezeket a  $T = (D; 0, 0, \dots, 0)$  virtuális tevékenységeket azután a hozzájuk tartozó közvetlen megelőzési relációk megadásával illeszthetjük az eredeti projektbe.

A 2.6 ábrán bemutatott globális pufferidő segítségével a projektenként *eltérő határidők* esete is kezelhető. Ehhez mindössze a szuperprojekt képzésekor kell az egyes projektek és a szuperprojekt záró tevékenysége közé egy – a fentiekhez hasonló - virtuális tevékenységet beilleszteni. Ha ezen virtuális tevékenységek hosszát alkalmasan  $T - T_i$ -nek választjuk meg (ahol  $T = \max_{1 \leq i \leq n} T_i$  a szuperprojekt,  $T_i$  pedig az  $i$ . projekt határideje), vagy kezdeteiket a megfelelő időpontban rögzítjük, akkor pontosan az eltérő határidők esetéhez jutunk.

A következő fejezetekben ismerttetendő modellek és megoldási módszerek az ilyen virtuális tevékenységek bevezetése mellett is változtatás nélkül alkalmazhatóak, így ezek tárgyalására a továbbiakban már nem térünk ki.



## ERŐFORRÁS KIEGYENLÍTÉSI MILP MODELLEK

A hagyományos erőforrás kiegyenlítő modelleknél legtöbbször alkalmazott kiegyenlítési mértékek (mint például az FA – az átlagos erőforrás kihasználtság körüli ingadozás, vagy az FB – az egymás utáni erőforrás felhasználás ingadozás) lényegében „lokális” mértékek. Ennek következtében alkalmatlanok arra, hogy az erőforrások felhasználását globális szempontból jellemezzék. Az ilyen lokális mértékek esetében az ideális erőforrás kihasználtsági diagram alakja téglalap. Ezt az ideális állapotot a projektek zömében képtelenség elérni, hiszen már a bevezetésben említett általános projektmenedzsment elméletek is azt igazolják, hogy a projekteknek „ívük” van, amely a projektalapítástól kezdődve felfelé, majd a projektzárásig lefelé ível. A kivitelezési fázisban az erőforrás kihasználtság pedig tele lesz csúcsokkal és hullámvölgyekkel, melyek a projektmenedzsment szemében tüskék, hiszen sokszori megszakításokat és gyakori állásidőket eredményeznek.

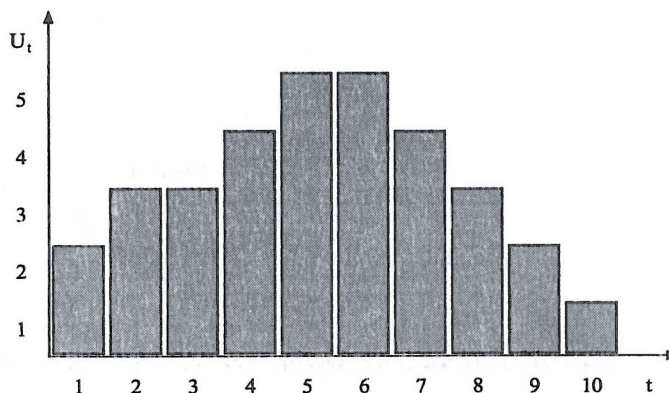
Az ismertetendő modell egyik legfőbb újdonsága, hogy olyan „globális” erőforrás kiegyenlítő mértéken alapszik, amely képes globálisan kezelni a fenti problémát. A modell alapját a gépütemezési problémák esetében már sikeresen alkalmazott módszerek és megoldási eljárások alkotják.

A gépütemezési feladatok esetében az állásidők (IT – Idle Times) játszanak központi szerepet. Azzal, hogy az új „globális” mérték tulajdonképpen a gépütemezésben ismert állásidők analógiája után készült, hasonló jó tulajdonságokat enged sejtetni, amelyek bizonyítására is sor kerül [5.3].

Első megközelítésben azt mondhatjuk, hogy egy adott időpillanatban egy erőforrás egység kihasználatlan (tétlen), ha egyetlen tevékenységhez sincs hozzárendelve. Egy ilyen erőforrás egység „aktivitása” igen változatos képet mutat-

hat. Természetes kíváncsi, hogy az erőforrás felhasználása folyamatos legyen, mely kívánság sok esetben nem teljesíthető, és így nem kívánt megszakításokhoz, tétlenségre ítélt időszakokhoz vezet. Ennek eredményeként számos erőforrás terhelése erősen ingadozó: várakozás, munkafázis, várakozás, munkafázis...

Felmerül a kérdés, hogy ha a „lokális” mértékek esetében a téglalap alakú erőforrás kihasználtsági diagram az ideális, akkor mi a helyzet a bevezetendő „globális” mértékünk esetén. A választ a kvázi-konkáv diagramok adják. A gyakorlati tapasztalatok azt mutatják, hogy a „globális” szempontból legjobban kezelhető erőforrás-kihasználtsági diagramja „hegyet” formáz, amely egy emelkedő, és egy csökkenő részből áll. Az ilyen erőforrás felhasználási profilok az erőforrás felhasználását egy centralizált blokkra koncentrálnak, jelentősen csökkentve a megszakításokat és az állásidőket, továbbá figyelembe veszik a beindítási és leállási folyamatok effektusait. Vegyük azonban észre, hogy a téglalap is egy kvázi-konkáv alakzat, tehát a modellünk által használni kívánt mérték egyfajta általánosítása az eddig használt, hagyományosan „lokális” mértékeknek. Az 5.1 ábrán egy tipikus kvázi-konkáv erőforrás-kihasználtsági diagramot láthatunk.



5.1 ábra: Egy tipikus kvázi-konkáv erőforrás kihasználtsági diagram egy erőforrás esetén

Az erőforrás kihasználtsági profil kvázi-konkávításának formális definíciója a következőképpen adható meg:

Minden  $t \in \{1, 2, \dots, T\}$  esetén jelölje  $U_t$  az  $S \in \mathcal{S}$  ütemezés által a  $t$ . időpontban felhasznált erőforrás mennyiségét.

**Definíció**

Az  $U = \{U_1, U_2, \dots, U_t, \dots, U_{T-1}, U_T\}$  erőforrás-kihasználtsági diagram kvázi konkáv, ha

$$\forall t \in \{2, 3, \dots, T-1\} : U_t \geq \min(U_{t-1}, U_{t+1}).$$

Ezen hagyományos definíció mellett szükségünk lesz egy ezzel egyenértékű definícióra is, mely az egyes időpontbeli erőforrás-felhasználások tágabb környezetein alapul. Ehhez segítségként vezessük be az  $U_t$  bal, illetve jobb oldali  $(MAXL_t, MAXR_t)$  maximális környezetét:

Legyen minden  $t \in \{1, 2, \dots, T\}$  esetén  $MAXL_t = \max(U_1, U_2, \dots, U_t)$ , és  $MAXR_t = \max(U_t, U_{t+1}, \dots, U_T)$ .

**Definíció**

Az  $U = \{U_1, U_2, \dots, U_t, \dots, U_{T-1}, U_T\}$  erőforrás-kihasználtsági diagram kvázi-konkáv, ha teljesülnek az alábbi egyenlőtlenségek:

$$\forall t \in \{2, 3, \dots, T-1\} : U_t \geq \min(MAXL_{t-1}, MAXR_{t+1}).$$

A továbbiakban igazoljuk is, hogy az előbbi két definíció használata felcserélhető.

**5.1 Lemma**

Az erőforrás diagramok hagyományos, illetve maximális környezetén alapuló definíciója ekvivalens.

**Bizonyítás:**

1. Először tegyük fel, hogy az  $U_t$  diagram kvázi-konkáv a maximális környezetén alapuló definíció szerint. Ekkor tudjuk, hogy  $\forall t \in \{2, 3, \dots, T-1\}$  esetén  $U_t \geq \min(MAXL_{t-1}, MAXR_{t+1})$ . Alkalmazva a maximális környezetek definícióját, illetve a maximum függvény monotonitását felhasználva, közvetlenül megkapjuk a hagyományos definíciót:



$$\begin{aligned}
U_t &\geq \min(MAXL_{t-1}, MAXR_{t+1}) \\
\min(MAXL_{t-1}, MAXR_{t+1}) &= \min(\max(U_{t-1}, MAXL_{t-2}), \max(U_{t+1}, MAXR_t)) \\
\min(\max(U_{t-1}, MAXL_{t-2}), \max(U_{t+1}, MAXR_t)) &\geq \min(U_{t-1}, U_{t+1})
\end{aligned}$$

2. Megfordítva pedig tegyük fel, hogy az  $U_t$  diagram kvázi-konkáv a hagyományos definíció szerint. Ekkor tudjuk, hogy  $\forall t \in \{2, 3, \dots, T-1\}$  esetén  $U_t \geq \min(U_{t-1}, U_{t+1})$ . Ha  $U_{t-1} \leq U_{t+1}$ , akkor az előbbi egyenlőtlenség  $U_t \geq U_{t-1}$  alakra hozható. A hagyományos definíciót csökkenő index szerint újra alkalmazva kapjuk, hogy  $U_t \geq U_{t-1} \geq \dots \geq U_1$ , vagyis  $U_t \geq MAXL_{t-1}$ , amiből az állítás egyből következik. Ha  $U_{t-1} \geq U_{t+1}$ , akkor hasonló módon igazolható, hogy  $U_t \geq U_{t+1} \geq \dots \geq U_T$ , illetve  $U_t \geq MAXR_{t+1}$ . A két részeredményt összegezve megállapíthatjuk, hogy  $U_t \geq \min(MAXL_{t-1}, MAXR_{t+1})$ . ♦

#### Definíció

Az  $\bar{U} = \{\bar{U}_1, \bar{U}_2, \dots, \bar{U}_t, \dots, \bar{U}_{T-1}, \bar{U}_T\}$  diagram az  $U$  erőforrás kihasználtsági profil kvázi-konkáv burkolója, ha:

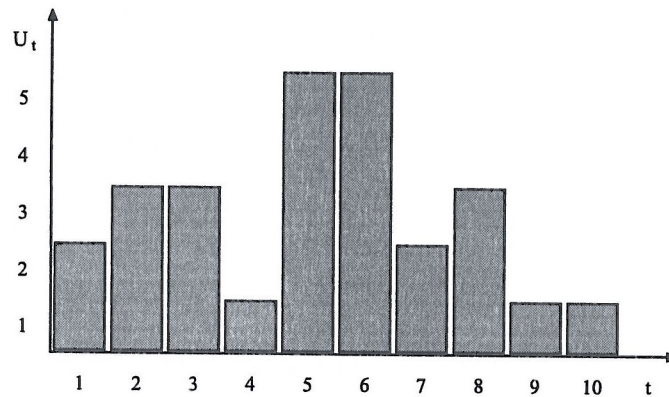
$$\bar{U}_1 = U_1$$

$$\bar{U}_t = \max(\min(MAXL_{t-1}, MAXR_{t+1}), U_t) \text{ ahol } t \in \{2, 3, \dots, T-1\}$$

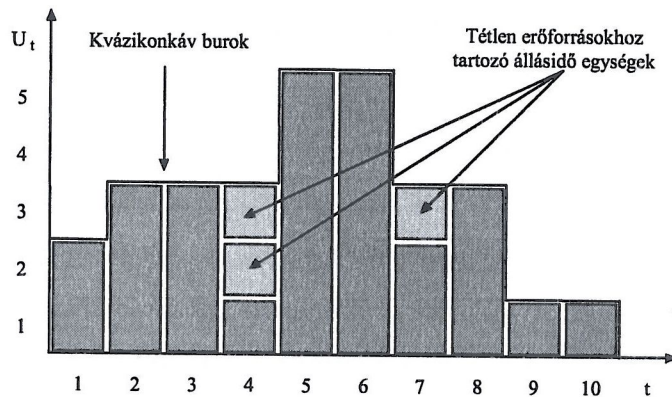
$$\bar{U}_T = U_T$$

Az 5.2 ábrán egy tipikus nem kvázi-konkáv erőforrás kihasználtsági diagramot láthatunk, melynek kvázi-konkáv burkolójában az 5.3 ábrán az állásidőkhöz tartozó egységeket is feltüntettük. Ebben a példában az erőforrás (gép- vagy embercsoport) folyamatos terhelés alatt van ugyan (nincsenek megszakításai), de a 4. időpillanatban az erőforrás 2. és 3. egysége, valamint a 7. időpillanatban a 3. egység újra tevékenység nélkül marad. Az összes állásidő mértéke jelen esetben 3 lesz.

Az állásidő mértékének számszerűsítésére többféle módszer alkalmazható. Ilyen például az összes megszakítások száma, vagy a folyamatosan megszakított erőforrás egységek száma, vagy a megszakításokból adódó (a diagram és annak kvázi-konkáv burkolója közötti terület különbség) terület nagysága. A szoftverfejlesztési projektek esetében ez utóbbi bizonyult használhatóbbnak, lévén a fejlesztői csapat tagjainak fenntartása minden egyes időpillanatban ugyanazzal a költséggel jár, akár van munkájuk, akár nincs. Így a továbbiakban ezt a változatot alkalmazzuk.



5.2 ábra: Egy tipikusan nem kvázi-konkáv erőforrás kihasználtsági diagram



5.3 ábra: Az erőforrás kihasználtság kvázi-konkáv burka és az állásidők

Ennek „globális alak orientált” megközelítésnek a hagyományos „lokális fluktuáció orientált” felfogással szemben az alábbi előnyei vannak:

1. Ha egy erőforrás kihasználtságának alakja kvázi-konkáv, akkor az állásidők a terhelési blokk elejére és végére koncentrálódnak, melyek jobban igazodnak a természetes alkalmazási folyamathoz, valamint többprojektes környezetben segítik a hatékonyabb felhasználást, a projektek között mozgathatóság javításával.

2. Az állásidő alapú új megközelítés bármennyire is hasonlít a hagyományos megközelítéshez, mégsem ugyanaz. Projektek esetében a menedzsment érdeke és feladata (mint azt az 1. fejezetben is említettük) valóban az adaptivitás és a rugalmasság fenntartása. Az is tény, hogy az erőforrások folyamatos, megszakítások nélküli felhasználása jelentősen csökkenti a projekt kivitelezésének költségét. Ez pedig pontosan az egyes erőforrás egységek felhasználásnak összes állásidejével jellemezhető, vagyis a kvázi-konkavitás a kulcsfogalom.

## 5.1 Globális erőforrás-kiegyenlítő mérték

Tekintsük tehát azt a globális erőforrás kiegyenlítési mértéket, amelyben a célfüggvény az egységnyi állásidők összegét tartalmazza. A kvázi-konkavitáson alapuló erőforrás kiegyenlítő IT mérték lényegét azon a ProMan 2.0 szoftverhez [5.4] készült *ESS* (ExtraSmallSize.net) demonstrációs projekten keresztül mutatjuk be, melyet az előző fejezet 4.2 és 4.3 ábráin már ismertettünk. A projekt mindössze kétféle erőforrást igényel, és 10 tevékenységből áll, melyet kiegészítettünk a nyitó és záró üres műveletekkel. Az egyes tevékenységek időigényét a téglalap hosszáról olvashatjuk le, míg az ábra többi értékének jelölése a következő alakú: *sorszám* [1. erőforrásigény, 2. erőforrásigény].

5.1 táblázat: Az ESS feladat leírás a  $T_i = (D_i; R_i^1, R_i^2)$  jelölésekkel

---

$P = \{T_0\} \cup \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}\} \cup \{T_{11}\}$			
$E = \{(T_0, T_1), (T_0, T_6), (T_1, T_2), (T_1, T_3), (T_2, T_4), (T_3, T_4), (T_4, T_5), (T_4, T_8)\}$			
$\cup \{(T_4, T_9), (T_5, T_7), (T_6, T_7), (T_7, T_{10}), (T_8, T_{11}), (T_9, T_{11}), (T_{10}, T_{11})\}$			
$T_0 = (1; 0, 0)$	$T_3 = (4; 3, 2)$	$T_6 = (5; 2, 1)$	$T_9 = (3; 1, 1)$
$T_1 = (2; 3, 3)$	$T_4 = (2; 4, 4)$	$T_7 = (3; 4, 3)$	$T_{10} = (2; 3, 2)$
$T_2 = (2; 3, 3)$	$T_5 = (3; 1, 1)$	$T_8 = (5; 2, 3)$	$T_{11} = (1; 0, 0)$

---



A modell összehasonlításához a globális IT mérték mellé a két leggyakrabban használt lokális mértéket - FA és FB – választottuk. A mértékek definiálásához jelöljük  $U_t^r(S) = U_t^r$ -vel az  $r$ . erőforrásból  $R \in \mathfrak{R}$  a  $t$ . időpontban  $t \in \{1, 2, \dots, T\}$  felhasznált mennyiséget az  $S \in \mathfrak{S}$  ütemezés mellett.

- Az IT (Idle Time) globális mérték szerinti célfüggvény a tétlen erőforrások állásidejének összegét írja le az  $r$ . erőforrásprofil, illetve annak kvázikonkáv burkolója közötti terület formájában

$$IT^r = \sum_{t=1}^T (\overline{U_t^r} - U_t^r)$$

- Az FA (Fluctuation around Average) lokális mérték célfüggvénye az  $r$ . erőforrás átlagos felhasználási szintjétől való eltérést méri:

$$FA^r = \sum_{t=1}^T (U_t^r - \frac{1}{T} \sum_{t=1}^T U_t^r)^2$$

- Az FB (Fluctuation Between periods) lokális mérték célfüggvénye pedig az  $r$ . erőforrás kihasználtságának egymás utáni ingadozását:

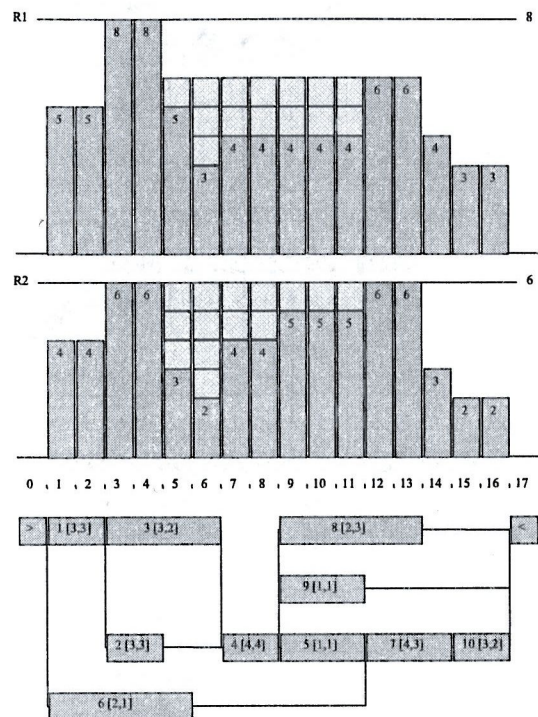
$$FB^r = \sum_{t=2}^T (U_t^r - U_{t-1}^r)^2$$

Az eredményeket az előbbi három mérték szerinti „*legjobb*” ütemezésekre, valamint az ES (legkorábbi ütemezés) esetében az 5.2 táblázatban foglaltuk össze. Az 5.4-5.7 ábrákon látható megoldások előállítására mindenütt az alábbi, erőforrásonként összegzett célfüggvényeket használtuk:

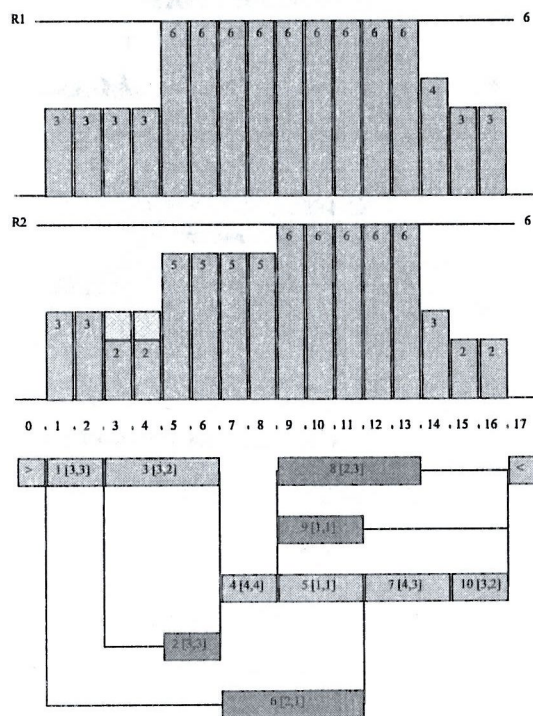
$$\sum_{r=1}^{|\mathfrak{R}|} M^r(S) \rightarrow \text{Min}, \text{ ahol } S \in \mathfrak{S} \text{ és } M \in \{IT, FA, FB\}$$

Az ütemezések előállítására a „brute force” nyers erő algoritmusát használtuk, lévén a lehetséges megoldások tere ennél a feladatnál meglehetősen kicsi  $|\mathfrak{S}_{\text{ESS}}| = 504$ .

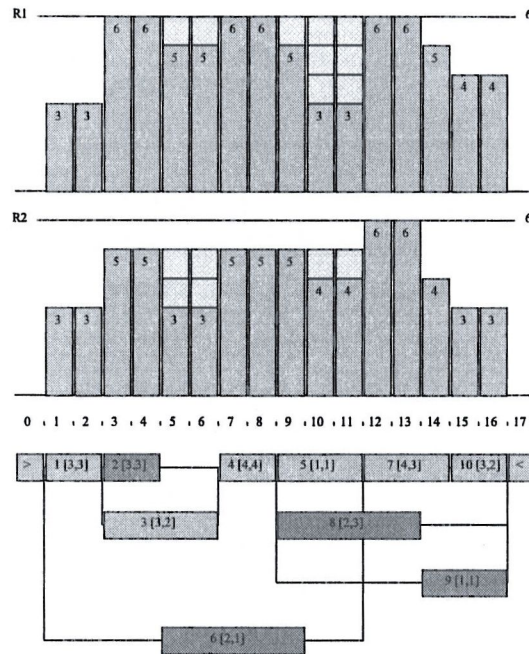
A megoldásokat a nem kritikus tevékenységek kezdési időpontjaival és az IT, FA, FB mértékek szerinti értékeivel jellemezzük. A Pareto optimumok közül mindegyik esetben a „legelsőnek” megtalált megoldást tekintjük „a” megoldásnak, lévén nagyobb méretű feladatok esetében is ez az általánosan elfogadott eljárás.



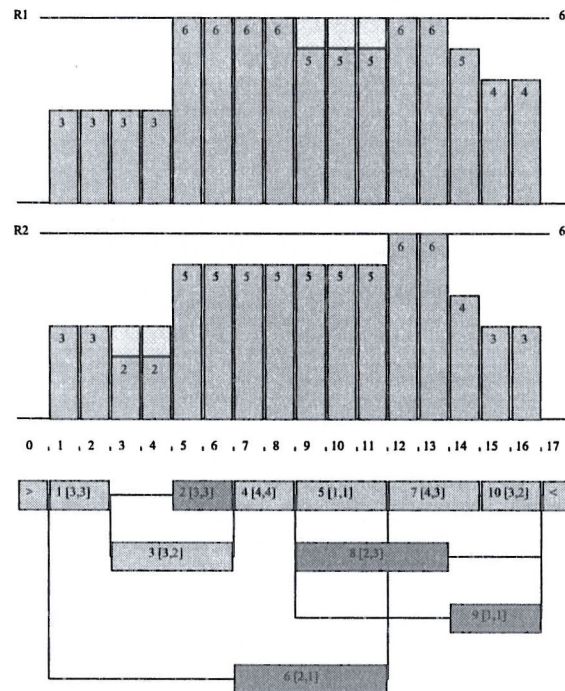
5.4 ábra: A legkorábbi kezdések (ES) által meghatározott ütemezés



5.5 ábra: A globális állásidők (IT) szerinti legjobb ütemezés



5.6 ábra: Az átlagtól való ingadozás (FA) mértéke szerint legjobb ütemezés



5.7 ábra: Az egymás utáni ingadozás (FB) mértéke szerint legjobb ütemezés



A futási eredményekről leolvashatjuk, hogy (1) az állásidőkön és kvázikonkavitáson alapuló mérték, illetve a hagyományos mértékek szerinti optimalizálás eltérő eredményeket ad - következésképpen alapjaiban különböznek. (2) Az ajánlott új, globális IT mérték segítségével valóban lehetséges az erőforrásprofilokkal szemben megfogalmazott egyenletességet (simaságot) elérni.

5.2 táblázat: Számítási eredmények összefoglalása

M	$\{IT_1, IT_2\}$	$\Sigma$	$\{FA_1, FA_2\}$	$\Sigma$	$\{FB_1, FB_2\}$	$\Sigma$	Hatékony ütemezések
ES	{14, 14}	28	{38, 33}	71	{32, 30}	62	$\{2 \Rightarrow 3, 8 \Rightarrow 9, 9 \Rightarrow 9, 6 \Rightarrow 1\}$
IT	{0, 2}	2	{34, 43}	77	{14, 21}	35	$\{2 \Rightarrow 5, 8 \Rightarrow 9, 9 \Rightarrow 9, 6 \Rightarrow 7\}$
FA	{9, 6}	15	<del>{24, 19}</del>	<del>43</del>	{27, 22}	49	$\{2 \Rightarrow 3, 8 \Rightarrow 9, 9 \Rightarrow 14, 6 \Rightarrow 5\}$
FB	{3, 2}	5	{24, 27}	51	<del>{13, 16}</del>	<del>29</del>	$\{2 \Rightarrow 5, 8 \Rightarrow 9, 9 \Rightarrow 14, 6 \Rightarrow 7\}$

## 5.2 Vegyes egészértékű programozási alap modell

Az alapmodell bemutatásakor már többféle erőforrást is megengedünk, de az egyes erőforrások között még nem teszünk különbséget. Az alapmodell megalkotásának első lépéseként a több kritérium szerinti optimalizálást kell helyettesíteni egyetlen kritériummal. Ennek egyik lehetséges módja az, ha új célfüggvényként az eredeti célfüggvények súlyozott összegét tekintjük. A modell leírásának szempontjából a súlyok semleges szerepet játszanak, ezért egységeknékné tekintjük őket.

Vezessük be a következő bináris kezdési időpontváltozókat, amelyek meghatározzák, hogy az egyes tevékenységek melyik időpontban kezdődnek:  $AS = \{AS_{it} | i = 1, 2, \dots, N; t = ES_i, \dots, LS_i\}$ . Az  $AS_{it} = 1$  akkor és csak akkor, ha az  $i$ . tevékenység végrehajtása a  $t$ . időpontban kezdődik, különben az értéke 0.

Az ütemezések során a tevékenységek nem bonthatóak meg, ezért mindegyik pontosan egyszer kezdődhet csak el. Képlettel megfogalmazva:  $\sum_{t=ES_i}^{LS_i} AS_{i,t} = 1$

és  $S_i = \sum_{t=ES_i}^{LS_i} t \cdot AS_{i,t}$  minden  $i = 1, 2, \dots, N$  esetén.

Az erőforráskorlátok mellőzése esetén a lehetséges megoldások a közvetlen megelőzési-rákövetkezési relációk úgynevezett „strong-x” leírásával adhatóak meg. [5.1]

$$\sum_{t=ES_i}^{LS_i} AS_{i,t} = 1 \text{ ahol } i = 1, 2, \dots, N$$

$$\sum_{s=t}^{LS_i} AS_{i,s} + \sum_{s=ES_j}^{t+D_j-1} AS_{j,s} \leq 1 \text{ ahol } t = ES_i, \dots, LS_i \text{ és } (i, j) \in E$$

A leírás első sora biztosítja azt, hogy minden tevékenység az  $[ES_i, LS_i]$  intervallumban pontosan egyszer kezdődjön meg. A második egyenlőtlenségben szereplő összeg értéke pedig csak akkor lehet 2, ha a  $j$ . tevékenység még az  $i$ . tevékenység befejezése előtt kezdődik el.

### Definíció

Az  $A$  mátrix *teljesen unimoduláris*, ha minden  $A_{i,j}$  eleme, továbbá minden aldeterminánsának értéke is +1, 0 vagy -1.

Bizonyítható, hogy ütemezési feladatok esetén az  $AS_{i,t}$  mátrix teljesen unimoduláris [5.5]. Ennek elméleti jelentőségét Chaudhuri [5.6] unimodularitási tétele adja, mely szerint ilyen esetekben az  $AS_{i,t} \in \{0, 1\}$  egészértékű változók helyett használhatunk  $0 \leq AS_{i,t} \leq 1$  valós változókat, és az így relaxált lineáris programozási (LP) feladat polinomiális időben megoldható. Sajnos esetünkben az erőforrások jelenléte miatt ez az út nem járható.

Tekintsük ezért most az  $S$  ütemezés  $t$ . időpontjában folyamatban lévő (aktív) tevékenységeinek  $\text{Act}(t) = \{i \in V \mid S_i \leq t < S_i + D_i\}$  halmazát. Ennek segítségével az  $U_i^r$  értéke már könnyen kifejezhető:

$$U_i^r = \sum_{i \in \text{Act}(t)} R_i^r = \sum_{i=1}^N \left( \sum_{j=t-D_i+1}^t AS_{i,j} \right) R_i^r$$

Megállapíthatjuk tehát, hogy az  $U_i^r$  értékek az  $AS_{i,t}$  kezdési időpont változótól lineárisan függenek.

A 4. fejezetben bevezetett jelölések analógiájára most is igyekszünk „beszédese” változóneveket alkalmazni. Legyenek tehát minden  $R \in \mathcal{R}$  erőforrásra, és  $t \in \{1, 2, \dots, T\}$  időpontra a bal, illetve jobb oldali maximális környezetek:

$$\text{MAXL}_i^r = \max(U_1^r, U_2^r, \dots, U_i^r) \text{ és } \text{MAXR}_i^r = \max(U_i^r, U_{i+1}^r, \dots, U_T^r).$$

A modellben használt állásidő mértéket úgy is kifejezhetjük, mint azon egysegnyi téglák száma, amelyekkel az erőforrás kihasználtsági diagram hullámvölgyeit fel kell töltenünk, hogy elérjük annak kvázi-konkáv burkát. Ez a verbális definíció az alábbi formulákkal fogalmazható meg egzakt módon:

$$IT^r(S) = \sum_{t=2}^{T-1} \max\left(\min(\text{MAXL}_{t-1}^r, \text{MAXR}_{t+1}^r) - U_t^r, 0\right)$$

Sajnos ennek a matematikai szempontból elegáns és „könnyen értelmezhető” globális mértéknek legalább két fontos negatív tulajdonsága is van:

(1) A minimum és a maximum függvények nem sima függvények. Ezért ha a linearitást meg akarjuk tartani, akkor ezeket a kifejezéseket lineáris korlátokkal vagy azok egészértékű változókkal kiegészített kombinációival kell helyettesítenünk.

(2) Az előbb említett helyettesítésekre a szélsőérték keresés iránya erős befolyással bír. Mivel a célfüggvény minimumát keressük, ezért a képletben szereplő minimum és maximum függvények esetén alkalmazott módszerek teljesen eltérhetnek egymástól. [5.2]



Legyen  $S \in \mathcal{S}$  egy lehetséges (a közvetlen megelőzési relációknak eleget tevő) ütemezés. A optimalizálási feladat ekkor az alábbi alakot ölti:

$$IT(S) \rightarrow \text{Min!}$$

$$S \in \mathcal{S}$$

A következő lépések sorozatával bebizonyítjuk, hogy alkalmas új változók és korlátozó feltételek bevezetésével ez a probléma linearizálható.

### 1. lépés

Először a számítási igény csökkentése érdekében a részletszámításokban rejlő duplikátumokat szüntetjük meg. A kettőnél több argumentummal rendelkező maximum függvényeket pontosan kétargumentumú maximum függvények sorozatával helyettesítjük.

$$MAXL_1' = U_1'$$

$$MAXL_t' = \max(MAXL_{t-1}', U_t') \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } R \in \mathcal{R}$$

$$MAXR_t' = \max(MAXR_{t+1}', U_t') \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } R \in \mathcal{R}$$

$$MAXR_T' = U_T'$$

### 2. lépés

Másodszor minden  $S \in \mathcal{S}$  lehetséges ütemezés esetén jelöljük  $I_t' = \bar{U}_t' - U_t'$ -vel a  $t \in \{2, 3, \dots, T-1\}$ -dik időpontban az  $r$ . erőforrásra vonatkozó állásidő egységek összegét, valamint vezessük be a  $MIN_t' = \min(MAXL_{t-1}', MAXR_{t+1}')$  rövidítést. Ekkor az első és második lépésből közvetlenül adódik, hogy eredeti probléma az alábbi nem sima optimalizálási alakra hozható:

$$\sum_{r=1}^k \sum_{t=2}^{T-1} I_t' \rightarrow \text{Min!}$$

$$U_t' + I_t' \geq MIN_t' \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MIN_t' = \min(MAXL_{t-1}', MAXR_{t+1}') \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$S \in \mathcal{S}$$

### 3. lépés

Mivel a célfüggvény minimumát keressük, az  $IT^r(S)$  függvények struktúrájából közvetlenül leolvasható, hogy

(1) A  $MAXL_t^r = \max(MAXL_{t-1}^r, U_t^r)$   $t \in \{2, 3, \dots, T-2\}$  egyenletek a következő lineáris korlátokkal helyettesíthetők:

$$MAXL_t^r \geq U_t^r$$

$$MAXL_t^r \geq MAXL_{t-1}^r.$$

(2) A  $MAXR_t^r = \max(MAXR_{t+1}^r, U_t^r)$   $t \in \{3, 4, \dots, T-1\}$  egyenletek esetén pedig a következő helyettesítések vezetnek célra:

$$MAXR_t^r \geq U_t^r$$

$$MAXR_t^r \geq MAXR_{t+1}^r$$

A harmadik lépés után a még mindig nem sima szélsőérték keresés a következő képletekkel fogalmazható meg:

$$\sum_{r=1}^k \sum_{t=2}^{T-1} I_t^r \rightarrow \text{Min!}$$

$$U_t^r + I_t^r \geq MIN_t^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MIN_t^r = \min(MAXL_{t-1}^r, MAXR_{t+1}^r) \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXL_1^r = U_1^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$MAXL_t^r \geq U_t^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXL_t^r \geq MAXL_{t-1}^r. \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_t^r \geq U_t^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_t^r \geq MAXR_{t+1}^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_T^r = U_T^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$S \in \mathcal{S}$$

#### 4. lépés

Minden egyes  $MIN_t^r = \min(MAXL_{t-1}^r, MAXR_{t+1}^r)$  egyenlet  $r \in \{1, \dots, k\}$  és  $t \in \{2, 3, \dots, T-1\}$  esetén helyettesíthető az alábbi nemlineáris korlátozó feltételekkel, ahol az  $LLTR_{t-1}^r$  bináris (0/1 értékű) változók. A bevezetett jelöléseknek megfelelően  $LLTR_{t-1}^r = 1$  pontosan akkor, ha  $MAXL_{t-1}^r > MAXR_{t+1}^r$ ; különben 0.

$$MIN_t^r = LLTR_{t-1}^r \cdot MAXL_{t-1}^r + (1 - LLTR_{t-1}^r) \cdot MAXR_{t+1}^r$$

$$MIN_t^r \leq MAXL_{t-1}^r \quad MIN_t^r \leq MAXR_{t+1}^r \quad LLTR_{t-1}^r \in \{0, 1\}$$

Az átírás azon egyszerű tényen alapul, hogy az  $M = \min(a, b)$  kifejezés egyenértékű az  $M = at + (1-t)b$ ,  $M \geq a$ ;  $M \geq b$ ,  $t \in \{0, 1\}$  feltételrendszerrel. A művelet után kapott rendszer teljes linearizálásához már csak az  $LLTR_{t-1}^r$  változót tartalmazó kételemű szorzatokat kell kiküszöbölni.

$$\sum_{r=1}^k \sum_{t=2}^{T-1} I_t^r \rightarrow \text{Min!}$$

$$U_t^r + I_t^r \geq MIN_t^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MIN_t^r \leq MAXL_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MIN_t^r \leq MAXR_{t+1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MIN_t^r = LLTR_{t-1}^r \cdot MAXL_{t-1}^r + (1 - LLTR_{t-1}^r) \cdot MAXR_{t+1}^r \\ \text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$LLTR_{t-1}^r \in \{0, 1\} \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXL_1^r = U_1^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$MAXL_t^r \geq U_t^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXL_t^r \geq MAXL_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_t^r \geq U_t^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_t^r \geq MAXR_{t+1}^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$MAXR_T^r = U_T^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$S \in \mathcal{S}$$



## 5. lépés

A "big-M" elnevezésű módszerrel [5.7] éppen ezeket a nemlineáris (pontosabban bilineáris  $LLTR_{t-1}^r \cdot MAXL_{t-1}^r$  és  $-LLTR_{t-1}^r \cdot MAXR_{t+1}^r$ ) korlátozó feltételeket válthatjuk ki az alábbi lineáris korlátokkal:

$$MIN_t^r \leq MAXL_{t-1}^r$$

$$MIN_t^r \leq MAXR_{t+1}^r$$

$$MIN_t^r = PROL_{t-1}^r + PROR_{t+1}^r$$

$$PROL_{t-1}^r \geq \underline{MAXL_{t-1}^r} \cdot LLTR_{t-1}^r$$

$$PROL_{t-1}^r \geq MAXL_{t-1}^r + \overline{MAXL_{t-1}^r} \cdot LLTR_{t-1}^r - \overline{MAXL_{t-1}^r}$$

$$PROR_{t+1}^r \geq -\overline{MAXR_{t+1}^r} \cdot LLTR_{t-1}^r + \underline{MAXR_{t+1}^r}$$

$$PROR_{t+1}^r \geq MAXR_{t+1}^r - \overline{MAXR_{t+1}^r} \cdot LLTR_{t-1}^r$$

$$LLTR_{t-1}^r \in \{0,1\}$$

A fenti képletekben a felül- illetve az alulvonással jelölt  $\underline{MAXL_{t-1}^r}$ ,  $\overline{MAXL_{t-1}^r}$ ,  $\underline{MAXR_{t+1}^r}$ , és  $\overline{MAXR_{t+1}^r}$  konstansok a megfelelő  $MAXL_{t-1}^r$  és  $MAXR_{t+1}^r$  értékek alsó, illetve felső korlátait jelölik, melyeket feladat paramétereiből számíthatunk ki.

## 6. lépés

Legutoljára az  $S \in \mathcal{S}$  feltételt helyettesítjük, annak „strong-x” leírásával.

$$\sum_{t=ES_i}^{LS_i} AS_{i,t} = 1 \text{ ahol } i = 1, 2, \dots, N$$

$$\sum_{s=t}^{LS_i} AS_{i,s} + \sum_{s=ES_j}^{t+D_j-1} AS_{j,s} \leq 1 \text{ ahol } t = ES_i, \dots, LS_i \text{ és } (i, j) \in E$$

$$AS_{i,t} \in \{0,1\} \text{ ahol } i = 1, 2, \dots, N \text{ és } t = ES_i, \dots, LS_i$$

$$U_t^r = \sum_{i=1}^N \left( \sum_{j=t-D_i+1}^t AS_{i,j} \right) R_i^r \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k\}$$

A hatodik lépés után az eredeti erőforrás kiegyenlítő problémát a következő vegyes egésztértékű lineáris programozási feladatra vezettük vissza [5.1]:

**IT (MILP – Mixed Integer Linear Programming) alapmodell**

$$\sum_{r=1}^k \sum_{t=2}^{T-1} I_t^r \rightarrow \text{Min!}$$

$$U_t^r + I_t^r \geq \text{MIN}_t^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MIN}_t^r \leq \text{MAXL}_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MIN}_t^r \leq \text{MAXR}_{t+1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MIN}_t^r = \text{PROL}_{t-1}^r + \text{PROR}_{t+1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{PROL}_{t-1}^r \geq \text{MAXL}_{t-1}^r \cdot \text{LLTR}_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{PROL}_{t-1}^r \geq \text{MAXL}_{t-1}^r + \overline{\text{MAXL}_{t-1}^r} \cdot \text{LLTR}_{t-1}^r - \overline{\text{MAXL}_{t-1}^r}$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{PROR}_{t+1}^r \geq -\overline{\text{MAXR}_{t+1}^r} \cdot \text{LLTR}_{t+1}^r + \overline{\text{MAXR}_{t+1}^r}$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{PROR}_{t+1}^r \geq \text{MAXR}_{t+1}^r - \overline{\text{MAXR}_{t+1}^r} \cdot \text{LLTR}_{t+1}^r$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MAXL}_1^r = U_1^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$\text{MAXL}_t^r \geq U_t^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MAXL}_t^r \geq \text{MAXL}_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MAXR}_t^r \geq U_t^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MAXR}_t^r \geq \text{MAXR}_{t+1}^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\text{MAXR}_T^r = U_T^r \text{ ahol } r \in \{1, 2, \dots, k\}$$

$$\text{LLTR}_{t-1}^r \in \{0,1\} \text{ ahol } t \in \{1, 2, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k\}$$

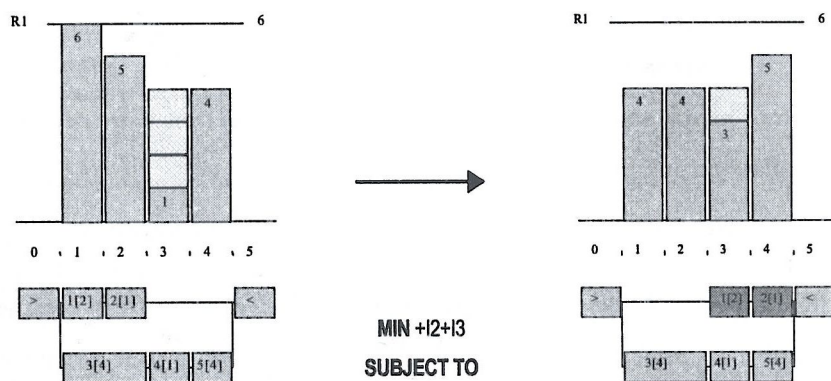
$$U_t^r = \sum_{i=1}^N \left( \sum_{j=t-D_i+1}^t \text{AS}_{i,j} \right) R_i^r \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k\}$$

$$\sum_{t=ES_i}^{LS_i} AS_{i,t} = 1 \text{ ahol } i = 1, 2, \dots, N$$

$$\sum_{s=t}^{LS_i} AS_{i,s} + \sum_{s=ES_j}^{t+D_j-1} AS_{j,s} \leq 1 \text{ ahol } t = ES_i, \dots, LS_i \text{ és } (i, j) \in E$$

$$AS_{i,t} \in \{0,1\} \text{ ahol } i = 1, 2, \dots, N \text{ és } t = ES_i, \dots, LS_i$$

A fenti IT *alapmodell* használatát egy kisméretű mintapéldán keresztül mutatjuk be. A feladat mindössze egy erőforrást, 2 projektet és 5 tevékenységet tartalmaz. A végső vegyes egésztértékű programozási feladatot a Lindo program jelölésszerint, periódusonként és feltétel csoportonként adtuk meg.



1. periódus	2. periódus	3. periódus	4. periódus
MAXL1-U1=0	+I2-MIN2+U2>=0	+I3-MIN3+U3>=0	MAXR4-U4=0
	+MAXL2-U2>=0	+MAXR3-U3>=0	
	-MAXL1+MAXL2>=0	+MAXR3-MAXR4>=0	
	+MIN2-MAXL1<=0	+MIN3-MAXL2<=0	
	+MIN2-MAXR3<=0	+MIN3-MAXR4<=0	
	+MIN2-PROL1-PROR3=0	+MIN3-PROL2-PROR4=0	
	-4LLTR1+PROL1>=0	-4LLTR2+PROL2>=0	
	-MAXL1-6LLTR1+PROL1>=-6	-MAXL2-6LLTR2+PROL2>=-6	
	+4LLTR1+PROR3>=4	+4LLTR2+PROR4>=4	
	-MAXR3+5LLTR1+PROR3>=0	-MAXR4+5LLTR2+PROR4>=0	
+2A1S1+4A3S1-U1=0	+2A1S2+A2S2+4A3S1-U2=0	+2A1S3+A2S3+A4S3-U3=0	A2S4+4A5S4-U4=0
	A1S1+A1S2+A1S3=1	A3S1=1	A1S2+A1S3+A2S2<=1
	A2S2+A2S3+A2S4=1	A4S3=1	A1S3+A2S2+A2S3<=1
		A5S4=1	
END			
INT A1S1 A1S2 A1S3 A2S2 A2S3 A2S4 A3S1 A4S3 A5S4 LLTR1 LLTR2			

5.8 ábra: Egy mintapélda IT modellje - legkorábbi és optimális ütemezése (Lindo)



### 5.3 Erősen és gyengén korlátos erőforrások MILP modellje

Végül rátérünk a teljes (gyengén és erősen korlátos erőforrásokat is kezelni tudó) modell leírására. Először megmutatjuk, hogy ez a probléma is modellezhető vegyes egészértékű programozási feladatként, melynek segítségével kisméretű projekteket könnyen megoldhatunk. Közepes méretű feladatok esetén a modell egzakt megoldására egy kétfázisú, implicit leszámláláson alapuló keresőfa eljárást adunk meg, mely hatékony metszési szabályaival lehetőséget ad az összes Pareto optimum meghatározására. Nagyméretű feladatok esetén, a probléma NP nehézsége révén heurisztikus megoldásokhoz kell folyamodnunk. A legkézenfekvőbb heurisztikákat ebben az esetben a közepes méretű feladatokra alkalmazott keresési fák mértének különféle korlátozásai adják.

Az illusztrációkhoz és a számítási eredmények közléséhez itt is a *ProMan 2.0* programot hívtuk segítségül, ahonnan egy kisméretű *SS* (SmallSize.net) problémát választottunk [5.4].

#### Definíció

Az  $R$  erőforrást *gyengén korlátos* erőforrásnak (röviden gyenge erőforrásnak) nevezzük, ha a projekt időtartama alatt nem adott az erőforráshoz tartozó  $R(t)$  rendelkezésre állási korlát. A gyengén korlátos erőforrások esetében az  $\mathcal{R}_s$  jelölést alkalmazzuk.

Valós helyzetekben természetesen minden erőforrásra megadható a rendelkezésre állási függvény (hiszen adott az alkalmazottak és gépek száma). Gyenge erőforrások esetében ezt nem vesszük figyelembe, hiszen az erőforrás jellegéből adódóan, ha az ütemezések során az adott erőforrásból hiány jelentkezik, akkor azt bármikor pótolni tudjuk pótlólagos erőforrások beszerzésével. A pótlólagos erőforrás származhat akár külső forrásból, akár belső forrásból. Ez utóbbi esetben ideiglenes munkaköri átcsoportosításokat kell alkalmazni.

#### Definíció

Az  $R$  erőforrást *erősen korlátos* erőforrásnak (röviden erős erőforrásnak) nevezzük, ha a projekt időtartama alatt adott erőforráshoz tartozó  $R(t)$  rendelkezés-

re állási korlát, és ezt egyetlen ütemezés sem lépheti túl, azaz  $U'_t \leq R(t)$ ,  $t = 1, 2, \dots, T$ . Az erősen korlátos erőforrások esetében az  $\mathcal{R}_H$  jelölést alkalmazzuk.

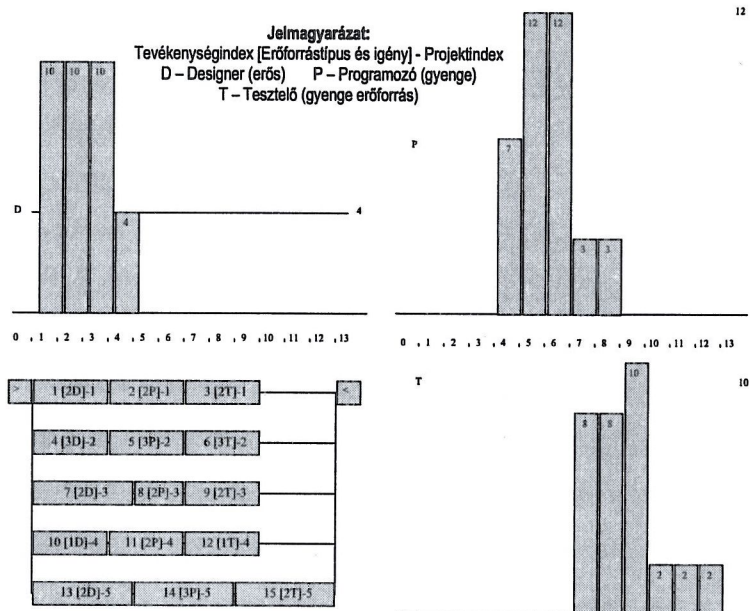
A 2.5 táblázatban már adtunk útmutatást arra, hogy miként lehet egy-egy erőforrás típusát (az adott projektre vonatkozóan) meghatározni. A fenti definíciók egy újabb, és objektívebb kategorizálási módszert adnak a kezünkbe. A pótlólagos erőforrások bevonásának költségei az erőforrás természetétől függetlenül a  $[0, \infty]$  intervallumban más és más értéket vehet fel. Az adott projekt szempontjából azonban meghatározhatunk egy olyan  $K$  küszöbértéket, mely alatt az adott erőforrás gyengének  $[0, K]$ , illetve felette erősnek minősíthető  $[K, \infty]$ . A rendelkezésre álló erőforrások tétlenségéből származó kieső idő költségére is hasonló megállapítás tehető.

### Definíció

Azokat az  $S \in \mathcal{S}$  ütemezéseket, amelyek nem okoznak ütközéseket az erős erőforrások felhasználása tekintetében, *elfogadható ütemezéseknek* nevezzük.

Megjegyezzük, hogy az elfogadható ütemezések egyben lehetséges ütemezések is. A két fogalom között éppen az (erős) erőforrás korlátok tesznek különbséget.

Az illusztrációként használt  $SS$  feladat paraméterei az 5.9 ábráról olvashatóak le. A feladat tulajdonképpen 5 egyszerű alprojektből képezett informatikai szuperprojekt. Alprojektenként az első tevékenység mindig a tervezés (design), amit a programozás majd végül a tesztelés követ. Minden tevékenység pontosan egyféle erőforrást igényel. Az erős erőforrásként definiált tervezők (D) létszáma a projekt időtartama alatt állandó (4 fő). Az ábráról az is látható, hogy a legkorábbi ES ütemezés ugyan lehetséges ütemezés, de nem elfogadható ütemezés, mert végrehajtásához nincs elegendő tervezőnk. A modell alapján kapott megoldásból (5.10 ábra) az is kiderül, hogy az erős erőforráskorlát figyelembe vételével a projekt minimális kivitelezési ideje 15 nap.



5.9 ábra: Az SS feladat leírása, a legkorábbi kezdés (ES) szerint ütemezés mellett

Az erősen korlátos erőforrások kezelése viszonylag könnyen modellezhető, hiszen az IT alapmodellhez mindössze néhány újabb (az erőforrás felhasználását) korlátozó feltételt kell hozzávennünk. Az egyszerűbb jelölés kedvéért feltételezzük, hogy az erős erőforrások mindig a legkisebb indexűek  $\mathcal{R}_H = \{R_1, R_2, \dots, R_k\}$ , illetve  $\mathcal{R}_S = \{R_{k+1}, R_{k+2}, \dots, R_{k+m}\}$ . Az erősen korlátos erőforrások mennyisége a projekt időtartama alatt konstans  $L' \equiv R^r(t)$   $r \in \{1, 2, \dots, k\}$ . Kisméretű vagy rövid távú projektek esetén ez amúgy sem jelent megszorítást. Az SS feladat esetében ez például  $L^1 \equiv 4$  fő, melyet az 5.9 ábra D erőforrásához tartozó diagramrészében meghúzott szintvonal jelez.

## 7. lépés

Az IT alapmodellt kiegészítve az alábbi korlátozó feltételekkel olyan modellt kapunk, melynek megoldása mindig elfogadható ütemezés.

$$U_i^r \leq R^r(t) \equiv L' \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k\}$$

Az erős erőforráskorlát definíciójából és az IT alapmodell tulajdonságaiból közvetlenül adódik, hogy a kiegészített IT modell lehetséges ütemezései éppen egybeesnek az eredeti alapmodell elfogadható ütemezéseivel.



## 8. lépés

De nem elégedhetünk meg csak az erős erőforrás-készleteink figyelembe vételével. Általában a gyenge erőforrások esetében is rendelkezünk saját készletekkel, és noha szükség esetén ezek mennyisége növelhető, a pótlólagos elemek bevonása csak szükség esetén preferált. Ezért az állásidők szerinti minimum megkeresésén túl, egyidejűleg az egyes gyenge erőforrásokra vonatkozó csúcsterheléseket is alacsony szinten kívánjuk tartani. (A „laposabb” kvázi-konkáv erőforrás-kihasználtsági diagramok egyenletesebb terhelést jelentenek.) Az  $S$  ütemezés által meghatározott, az  $R \in \mathcal{R}_S$  erőforrásra vonatkozó maximális kihasználási (MU – Maximal Usage) szintet a következő függvénnyel adhatjuk meg:

$$MU^r(S) = \max(U_1^r, U_2^r, \dots, U_T^r) = \max(MAXL_t^r, MAXR_t^r)$$

Az előbbi kiegészített modellben a kétféle (IT és MU) mérték szerinti egyidejű optimalizálási feladatot a célfüggvények (súlyozott) összegével írhatjuk le. Az  $r \in \{k+1, k+2, \dots, k+m\}$  gyenge erőforrások mindegyikére egy új, globális  $M^r$  felsőkorlát változót vezetünk be, amelyek a célfüggvényben az  $MU^r$  szinteket hivatottak minimalizálni. Ekkor a végső modell az alábbi alakban írható fel:

**Gyenge és erős erőforrásokkal kiegészített  $IT_{SH}$  (MILP) modell**

**Célfüggvény:**

$$\sum_{r=1}^{k+m} \sum_{t=2}^{T-1} I_t^r + \sum_{r=k+1}^{k+m} M^r \rightarrow \text{Min!} \quad (C1)$$

**Feltételrendszer:**

$$U_t^r \leq R(t) \equiv L^r \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k\} \quad (F1a)$$

$$U_t^r \leq M^r \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{k+1, k+2, \dots, k+m\} \quad (F1b)$$

$$U_t^r + I_t^r \geq MIN_t^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F2)$$

$$MIN_t^r \leq MAXL_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F3)$$

$$MIN_t^r \leq MAXR_{t+1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F4)$$

$$MIN_t^r = PROL_{t-1}^r + PROR_{t+1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F5)$$

$$PROL_{t-1}^r \geq \underline{MAXL}_{t-1}^r \cdot LLTR_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F6)$$

$$PROL_{t-1}^r \geq MAXL_{t-1}^r + \overline{MAXL}_{t-1}^r \cdot LLTR_{t-1}^r - \overline{MAXL}_{t-1}^r \quad (F7)$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\}$$

$$PROR_{t+1}^r \geq -\underline{MAXR}_{t+1}^r \cdot LLTR_{t-1}^r + \underline{MAXR}_{t+1}^r \quad (F8)$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\}$$

$$PROR_{t+1}^r \geq MAXR_{t+1}^r - \overline{MAXR}_{t+1}^r \cdot LLTR_{t-1}^r \quad (F9)$$

$$\text{ahol } t \in \{2, 3, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\}$$

$$MAXL_1^r = U_1^r \text{ ahol } r \in \{1, 2, \dots, k+m\} \quad (F10)$$

$$MAXL_t^r \geq U_t^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F11)$$

$$MAXL_t^r \geq MAXL_{t-1}^r \text{ ahol } t \in \{2, 3, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F12)$$

$$MAXR_t^r \geq U_t^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F13)$$

$$MAXR_t^r \geq MAXR_{t+1}^r \text{ ahol } t \in \{3, 4, \dots, T-1\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F14)$$

$$MAXR_T^r = U_T^r \text{ ahol } r \in \{1, 2, \dots, k+m\} \quad (F15)$$

$$LLTR_{t-1}^r \in \{0,1\} \text{ ahol } t \in \{1, 2, \dots, T-2\} \text{ és } r \in \{1, 2, \dots, k+m\} \quad (F16)$$

$$\sum_{t=ES_i}^{LS_i} AS_{it} = 1 \text{ ahol } i = 1, 2, \dots, N \quad (F17)$$

$$U_t^r = \sum_{i=1}^N \left( \sum_{j=t-D_i+1}^t AS_{i,j} \right) R_i^r \quad (F18)$$

$$\text{ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k+m\}$$

$$\sum_{s=t}^{LS_i} AS_{is} + \sum_{s=ES_j}^{t+D_i-1} AS_{js} \leq 1 \text{ ahol } t = ES_i, \dots, LS_i \text{ és } i \rightarrow j \in E \quad (F19)$$

$$AS_{it} \in \{0,1\} \text{ ahol } i = 1, 2, \dots, N \text{ és } t = ES_i, \dots, LS_i \quad (F20)$$

Mivel az  $IT_{SH}$  modellhez vezető lépések mind algoritmizálhatóak, ezért készítettünk egy olyan segédprogramot, amely az eredeti formában megadott erőforrás kiegyenlítési probléma alapparamétereiből képes meghatározni a fenti modellben szereplő konstansokat, és ezek segítségével megadja a hozzá tartozó  $IT_{SH}$  MILP feladatot. Az itt közreadandó  $SS$  feladathoz tartozó, vegyes egészcértékű lineáris programozási *Lindo* modell is ezzel készült.

A program egységes jelölésrendszerét megtartottuk, vagyis  $D=R1$  (designer) erős erőforrás, és  $P=R2$ ,  $T=R3$  (programozó és tesztelő) gyenge erőforrások:

! Hard Leveling and Maximal Usage and Idle Time

MIN

! (C1) célfüggvény

$+R1I2+R1I3+R1I4+R1I5+R1I6+R1I7+R1I8+R1I9+R1I10+R1I11+R1I12+R1I13+R1I14$ $+R2I2+R2I3+R2I4+R2I5+R2I6+R2I7+R2I8+R2I9+R2I10+R2I11+R2I12+R2I13+R2I14$ $+R3I2+R3I3+R3I4+R3I5+R3I6+R3I7+R3I8+R3I9+R3I10+R3I11+R3I12+R3I13+R3I14$ $+M2$ $+M3$
--

SUBJECT TO

! (F1a) és (F1b) típusú feltételek

$R1U1 \leq 4$	$R2U1-M2 \leq 0$	$R3U1-M3 \leq 0$
$R1U2 \leq 4$	$R2U2-M2 \leq 0$	$R3U2-M3 \leq 0$
...	...	...
$R1U15 \leq 4$	$R2U15-M2 \leq 0$	$R3U15-M3 \leq 0$

! (F2) típusú feltételek

$R1I2-R1MIN2+R1U2 \geq 0$	$R2I2-R2MIN2+R2U2 \geq 0$	$R3I2-R3MIN2+R3U2 \geq 0$
$R1I3-R1MIN3+R1U3 \geq 0$	$R2I3-R2MIN3+R2U3 \geq 0$	$R3I3-R3MIN3+R3U3 \geq 0$
...	...	...
$R1I14-R1MIN14+R1U14 \geq 0$	$R2I14-R2MIN14+R2U14 \geq 0$	$R3I14-R3MIN14+R3U14 \geq 0$

! (F3) típusú feltételek

$R1MIN2-R1MAXL1 \leq 0$	$R2MIN2-R2MAXL1 \leq 0$	$R3MIN2-R3MAXL1 \leq 0$
$R1MIN3-R1MAXL2 \leq 0$	$R2MIN3-R2MAXL2 \leq 0$	$R3MIN3-R3MAXL2 \leq 0$
...	...	...
$R1MIN14-R1MAXL13 \leq 0$	$R2MIN14-R2MAXL13 \leq 0$	$R3MIN14-R3MAXL13 \leq 0$

! (F4) típusú feltételek

$R1MIN2-R1MAXR3 \leq 0$	$R2MIN2-R2MAXR3 \leq 0$	$R3MIN2-R3MAXR3 \leq 0$
$R1MIN3-R1MAXR4 \leq 0$	$R2MIN3-R2MAXR4 \leq 0$	$R3MIN3-R3MAXR4 \leq 0$
...	...	...
$R1MIN14-R1MAXR15 \leq 0$	$R2MIN14-R2MAXR15 \leq 0$	$R3MIN14-R3MAXR15 \leq 0$

! (F5) típusú feltételek

$R1MIN2-R1PROL1-R1PROR3=0$	$R2MIN2-R2PROL1-R2PROR3=0$	$R3MIN2-R3PROL1-R3PROR3=0$
$R1MIN3-R1PROL2-R1PROR4=0$	$R2MIN3-R2PROL2-R2PROR4=0$	$R3MIN3-R3PROL2-R3PROR4=0$
...	...	...
$R1MIN14-R1PROL13-R1PROR15=0$	$R2MIN14-R2PROL13-R2PROR15=0$	$R3MIN14-R3PROL13-R3PROR15=0$

! (F6) típusú feltételek

$R1PROL1 \geq 0$	$R2PROL1 \geq 0$	$R3PROL1 \geq 0$
$R1PROL2 \geq 0$	$R2PROL2 \geq 0$	$R3PROL2 \geq 0$
...	...	...
$R1PROL3 \geq 0$	$R2PROL7 \geq 0$	$R3PROL11 \geq 0$
$-2R1LLTR4+R1PROL4 \geq 0$	$-3R2LLTR8+R2PROL8 \geq 0$	$-2R3LLTR12+R3PROL12 \geq 0$
$-2R1LLTR5+R1PROL5 \geq 0$	$-3R2LLTR9+R2PROL9 \geq 0$	$-2R3LLTR13+R3PROL13 \geq 0$
...	...	...
$-2R1LLTR13+R1PROL13 \geq 0$	$-3R2LLTR13+R2PROL13 \geq 0$	

! (F7) típusú feltételek

	$-R2MAXL1+R2PROL1 \geq 0$ $-R2MAXL2+R2PROL2 \geq 0$ $-R2MAXL3+R2PROL3 \geq 0$	$-R3MAXL1+R3PROL1 \geq 0$ $-R3MAXL2+R3PROL2 \geq 0$ ... $-R3MAXL6+R3PROL6 \geq 0$
	$-R2MAXL4-7R2LLTR4+R2PROL4 \geq -7$	$-R3MAXL7-8R3LLTR7+R3PROL7 \geq -8$ $-R3MAXL8-8R3LLTR8+R3PROL8 \geq -8$
$-R1MAXL2-10R1LLTR2+R1PROL2 \geq -10$ $-R1MAXL3-10R1LLTR3+R1PROL3 \geq -10$ ...	$-R2MAXL5-12R2LLTR5+R2PROL5 \geq -12$ $-R2MAXL6-12R2LLTR6+R2PROL6 \geq -12$ ...	$-R3MAXL9-10R3LLTR9+R3PROL9 \geq -10$ $-R3MAXL10-10R3LLTR10+R3PROL10 \geq -10$ ...
$-R1MAXL13-10R1LLTR13+R1PROL13 \geq -10$	$-R2MAXL13-12R2LLTR13+R2PROL13 \geq -12$	$-R3MAXL13-10R3LLTR13+R3PROL13 \geq -10$

! (F8) típusú feltételek

$R1PROR5 \geq 0$	$R2PROR9 \geq 0$	$R3PROR13 \geq 0$
$R1PROR6 \geq 0$	$R2PROR10 \geq 0$	...
...	...	$R3PROR14 \geq 0$
$R1PROR15 \geq 0$	$R2PROR15 \geq 0$	$R3PROR15 \geq 0$
$+2R1LLTR1+R1PROR3 \geq 2$ $+2R1LLTR2+R1PROR4 \geq 2$	$+3R2LLTR1+R2PROR3 \geq 3$ $+3R2LLTR2+R2PROR4 \geq 3$ ...	$+2R3LLTR1+R3PROR3 \geq 2$ $+2R3LLTR2+R3PROR4 \geq 2$ ...
	$+3R2LLTR6+R2PROR8 \geq 3$	$+2R3LLTR10+R3PROR12 \geq 2$



! (F9) típusú feltételek

-R1MAXR3+10R1LLTR1+R1PROR3>=0 -R1MAXR4+10R1LLTR2+R1PROR4>=0 ... -R1MAXR10+2R1LLTR8+R1PROR10>=0	-R2MAXR3+12R2LLTR1+R2PROR3>=0 -R2MAXR4+12R2LLTR2+R2PROR4>=0 ... -R2MAXR12+9R2LLTR10+R2PROR12>=0	-R3MAXR3+10R3LLTR1+R3PROR3>=0 -R3MAXR4+10R3LLTR2+R3PROR4>=0 ... -R3MAXR15+10R3LLTR13+R3PROR15>=0
-R1MAXR11+R1PROR11>=0 -R1MAXR12+R1PROR12>=0 ... -R1MAXR15+R1PROR15>=0	-R2MAXR13+R2PROR13>=0 -R2MAXR14+R2PROR14>=0 -R2MAXR15+R2PROR15>=0	

! (F10) típusú feltételek

R1MAXL1-R1U1=0	R2MAXL1-R2U1=0	R3MAXL1-R3U1=0
----------------	----------------	----------------

! (F11) típusú feltételek

R1MAXL2-R1U2>=0 R1MAXL3-R1U3>=0 ... R1MAXL13-R1U13>=0	R2MAXL2-R2U2>=0 R2MAXL3-R2U3>=0 ... R2MAXL13-R2U13>=0	R3MAXL2-R3U2>=0 R3MAXL3-R3U3>=0 ... R3MAXL13-R3U13>=0
--	--	--

! (F12) típusú feltételek

-R1MAXL1+R1MAXL2>=0 -R1MAXL2+R1MAXL3>=0 ... -R1MAXL12+R1MAXL13>=0	-R2MAXL1+R2MAXL2>=0 -R2MAXL2+R2MAXL3>=0 ... -R2MAXL12+R2MAXL13>=0	-R3MAXL1+R3MAXL2>=0 -R3MAXL2+R3MAXL3>=0 ... -R3MAXL12+R3MAXL13>=0
--	--	--

! (F13) típusú feltételek

R1MAXR3-R1U3>=0 R1MAXR4-R1U4>=0 ... R1MAXR14-R1U14>=0	R2MAXR3-R2U3>=0 R2MAXR4-R2U4>=0 ... R2MAXR14-R2U14>=0	R3MAXR3-R3U3>=0 R3MAXR4-R3U4>=0 ... R3MAXR14-R3U14>=0
--	--	--

! (F14) típusú feltételek

R1MAXR3-R1MAXR4>=0 R1MAXR4-R1MAXR5>=0 ... R1MAXR14-R1MAXR15>=0	R2MAXR3-R2MAXR4>=0 R2MAXR4-R2MAXR5>=0 ... R2MAXR14-R2MAXR15>=0	R3MAXR3-R3MAXR4>=0 R3MAXR4-R3MAXR5>=0 ... R3MAXR14-R3MAXR15>=0
---	---	---

! (F15) típusú feltételek

R1MAXR15-R1U15=0	R2MAXR15-R2U15=0	R3MAXR15-R3U15=0
------------------	------------------	------------------

! (F17) típusú feltételek

A1S1+A1S2+...+A1S7=1 A2S4+A2S5+...+A2S10=1 A3S7+A3S8+...+A3S13=1 A4S1+A4S2+...+A4S7=1 A5S4+A5S5+...+A5S10=1	A6S7+A6S8+...+A6S13=1 A7S1+A7S2+...+A7S7=1 A8S5+A8S6+...+A8S11=1 A9S7+A9S8+...+A9S13=1 A10S1+A10S2+...+A10S7=1	A11S4+A11S5+...+A11S10=1 A12S7+A12S8+...+A12S13=1 A13S1+A13S2+...+A13S4=1 A14S5+A14S6+...+A14S8=1 A15S9+A15S10+...+A15S12=1
---	--	---

! (F18) típusú feltételek

0+=2A1S1+3A4S1+2A7S1 +A10S1+2A13S1-R1U1 0+=2A1S1+2A1S2+3A4S1 +3A4S2+2A7S1+2A7S2 +A10S1+A10S2+2A13S1 +2A13S2-R1U2 0+=2A1S1+2A1S2+2A1S3 +3A4S1+3A4S2+3A4S3 +2A7S1+2A7S2+2A7S3 +A10S1+A10S2+A10S3 +2A13S1+2A13S2+2A13S3 -R1U3 0+=2A1S2+2A1S3+2A1S4 +3A4S2+3A4S3+3A4S4 +2A7S1+2A7S2+2A7S3 +2A7S4+A10S2+A10S3 +A10S4+2A13S1+2A13S2 +2A13S3+2A13S4-R1U4 0+=2A1S3+2A1S4+2A1S5 +3A4S3+3A4S4+3A4S5 +2A7S2+2A7S3+2A7S4 +2A7S5+A10S3+A10S4 +A10S5+2A13S2+2A13S3 +2A13S4-R1U5 0+=2A1S4+2A1S5+2A1S6 +3A4S4+3A4S5+3A4S6 +2A7S3+2A7S4+2A7S5 +2A7S6+A10S4+A10S5 +A10S6+2A13S3+2A13S4	0=-R2U1 0=-R2U2 0=-R2U3 0+=2A2S4+3A5S4+2A11S4 -R2U4 0+=2A2S4+2A2S5+3A5S4 +3A5S5+2A8S5+2A11S4 +2A11S5+3A14S5-R2U5 0+=2A2S4+2A2S5+2A2S6 +3A5S4+3A5S5+3A5S6 +2A8S5+2A8S6+2A11S4 +2A11S5+2A11S6+3A14S5 +3A14S6-R2U6 0+=2A2S5+2A2S6+2A2S7 +3A5S5+3A5S6+3A5S7 +2A8S6+2A8S7+2A11S5 +2A11S6+2A11S7+3A14S5 +3A14S6+3A14S7-R2U7 0+=2A2S6+2A2S7+2A2S8 +3A5S6+3A5S7+3A5S8 +2A8S7+2A8S8+2A11S6 +2A11S7+2A11S8+3A14S5 +3A14S6+3A14S7+3A14S8 -R2U8 0+=2A2S7+2A2S8+2A2S9 +3A5S7+3A5S8+3A5S9 +2A8S8+2A8S9+2A11S7 +2A11S8+2A11S9+3A14S6 +3A14S7+3A14S8-R2U9	0=-R3U1 0=-R3U2 0=-R3U3 0=-R3U4 0=-R3U5 0=-R3U6 0+=2A3S7+3A6S7+2A9S7 +A12S7-R3U7 0+=2A3S7+2A3S8+3A6S7 +3A6S8+2A9S7+2A9S8 +A12S7+A12S8-R3U8 0+=2A3S7+2A3S8+2A3S9 +3A6S7+3A6S8+3A6S9 +2A9S7+2A9S8+2A9S9 +A12S7+A12S8+A12S9 +2A15S9-R3U9 0+=2A3S8+2A3S9+2A3S10 +3A6S8+3A6S9+3A6S10 +2A9S8+2A9S9+2A9S10 +A12S8+A12S9+A12S10 +2A15S9+2A15S10-R3U10 0+=2A3S9+2A3S10+2A3S11 +3A6S9+3A6S10+3A6S11 +2A9S9+2A9S10+2A9S11 +A12S9+A12S10+A12S11 +2A15S9+2A15S10+2A15S11 -R3U11 0+=2A3S10+2A3S11+2A3S12 +3A6S10+3A6S11+3A6S12
--	---	--

-R1U6 0=+2A1S5+2A1S6+2A1S7 +3A4S5+3A4S6+3A4S7 +2A7S4+2A7S5+2A7S6 +2A7S7+A10S5+A10S6 +A10S7+2A13S4-R1U7 0=+2A1S6+2A1S7+3A4S6 +3A4S7+2A7S5+2A7S6 +2A7S7+A10S6+A10S7-R1U8 0=+2A1S7+3A4S7+2A7S6 +2A7S7+A10S7-R1U9 0=+2A7S7-R1U10 0=-R1U11 0=-R1U12 0=-R1U13 0=-R1U14 0=-R1U15	0=+2A2S8+2A2S9+2A2S10 +3A5S8+3A5S9+3A5S10 +2A8S9+2A8S10+2A11S8 +2A11S9+2A11S10+3A14S7 +3A14S8-R2U10 0=+2A2S9+2A2S10+3A5S9 +3A5S10+2A8S10+2A8S11 +2A11S9+2A11S10+3A14S8 -R2U11 0=+2A2S10+3A5S10+2A8S11 +2A11S10-R2U12 0=-R2U13 0=-R2U14 0=-R2U15	+2A9S10+2A9S11+2A9S12 +A12S10+A12S11+A12S12 +2A15S9+2A15S10+2A15S11 +2A15S12-R3U12 0=+2A3S11+2A3S12+2A3S13 +3A6S11+3A6S12+3A6S13 +2A9S11+2A9S12+2A9S13 +A12S11+A12S12+A12S13 +2A15S10+2A15S11+2A15S12 -R3U13 0=+2A3S12+2A3S13+3A6S12 +3A6S13+2A9S12+2A9S13 +A12S12+A12S13+2A15S11 +2A15S12-R3U14 0=+2A3S13+3A6S13+2A9S13 +A12S13+2A15S12 -R3U15
---	--	---

! (F19) típusú feltételek

(A1S2+A1S3+...+A1S7)+(A2S4)<=1 (A1S3+A1S4+...+A1S7)+(A2S4+A2S5)<=1 (A1S4+A1S5+...+A1S7)+(A2S4+A2S5+A2S6)<=1 (A1S5+A1S6+A1S7)+(A2S4+A2S5+...+A2S7)<=1 (A1S6+A1S7)+(A2S4+A2S5+...+A2S8)<=1 (A1S7)+(A2S4+A2S5+...+A2S9)<=1	(A2S5+A2S6+...+A2S10)+(A3S7)<=1 (A2S6+A2S7+...+A2S10)+(A3S7+A3S8)<=1 (A2S7+A2S8+...+A2S10)+(A3S7+A3S8+A3S9)<=1 (A2S8+A2S9+A2S10)+(A3S7+A3S8+...+A3S10)<=1 (A2S9+A2S10)+(A3S7+A3S8+...+A3S11)<=1 (A2S10)+(A3S7+A3S8+...+A3S12)<=1
(A4S2+A4S3+...+A4S7)+(A5S4)<=1 (A4S3+A4S4+...+A4S7)+(A5S4+A5S5)<=1 (A4S4+A4S5+...+A4S7)+(A5S4+A5S5+A5S6)<=1 (A4S5+A4S6+A4S7)+(A5S4+A5S5+...+A5S7)<=1 (A4S6+A4S7)+(A5S4+A5S5+...+A5S8)<=1 (A4S7)+(A5S4+A5S5+...+A5S9)<=1	(A5S5+A5S6+...+A5S10)+(A6S7)<=1 (A5S6+A5S7+...+A5S10)+(A6S7+A6S8)<=1 (A5S7+A5S8+...+A5S10)+(A6S7+A6S8+A6S9)<=1 (A5S8+A5S9+A5S10)+(A6S7+A6S8+...+A6S10)<=1 (A5S9+A5S10)+(A6S7+A6S8+...+A6S11)<=1 (A5S10)+(A6S7+A6S8+...+A6S12)<=1
(A7S2+A7S3+...+A7S7)+(A8S5)<=1 (A7S3+A7S4+...+A7S7)+(A8S5+A8S6)<=1 (A7S4+A7S5+...+A7S7)+(A8S5+A8S6+A8S7)<=1 (A7S5+A7S6+A7S7)+(A8S5+A8S6+...+A8S8)<=1 (A7S6+A7S7)+(A8S5+A8S6+...+A8S9)<=1 (A7S7)+(A8S5+A8S6+...+A8S10)<=1	(A8S6+A8S7+...+A8S11)+(A9S7)<=1 (A8S7+A8S8+...+A8S11)+(A9S7+A9S8)<=1 (A8S8+A8S9+...+A8S11)+(A9S7+A9S8+A9S9)<=1 (A8S9+A8S10+A8S11)+(A9S7+A9S8+...+A9S10)<=1 (A8S10+A8S11)+(A9S7+A9S8+...+A9S11)<=1 (A8S11)+(A9S7+A9S8+...+A9S12)<=1
(A10S2+A10S3+...+A10S7)+(A11S4)<=1 (A10S3+A10S4+...+A10S7)+(A11S4+A11S5)<=1 (A10S4+A10S5+...+A10S7)+(A11S4+A11S5+A11S6)<=1 (A10S5+A10S6+A10S7)+(A11S4+A11S5+...+A11S7)<=1 (A10S6+A10S7)+(A11S4+A11S5+...+A11S8)<=1 (A10S7)+(A11S4+A11S5+...+A11S9)<=1	(A11S5+A11S6+...+A11S10)+(A12S7)<=1 (A11S6+A11S7+...+A11S10)+(A12S7+A12S8)<=1 (A11S7+A11S8+...+A11S10)+(A12S7+A12S8+A12S9)<=1 (A11S8+A11S9+A11S10)+(A12S7+A12S8+...+A12S10)<=1 (A11S9+A11S10)+(A12S7+A12S8+...+A12S11)<=1 (A11S10)+(A12S7+A12S8+...+A12S12)<=1
(A13S2+A13S3+A13S4)+(A14S5)<=1 (A13S3+A13S4)+(A14S5+A14S6)<=1 (A13S4)+(A14S5+A14S6+A14S7)<=1	(A14S6+A14S7+A14S8)+(A15S9)<=1 (A14S7+A14S8)+(A15S9+A15S10)<=1 (A14S8+A15S9+A15S10)+(A15S11)<=1

END

! (F16) típusú feltételek

INT R1LLTR1	INT R2LLTR1	INT R3LLTR1
INT R1LLTR2	INT R2LLTR2	INT R3LLTR2
...	...	...
INT R1LLTR13	INT R2LLTR13	INT R3LLTR13

! (F20) típusú feltételek

INT A1S1	INT A2S4	INT A3S7	INT A4S1	INT A5S4
INT A1S2	INT A2S5	INT A3S8	INT A4S2	INT A5S5
...	...	...	...	...
INT A1S7	INT A2S10	INT A3S13	INT A4S7	INT A5S10
INT A6S7	INT A7S1	INT A8S5	INT A9S7	INT A10S1
INT A6S8	INT A7S2	INT A8S6	INT A9S8	INT A10S2
...	...	...	...	...
INT A6S13	INT A7S7	INT A8S11	INT A9S13	INT A10S7
INT A11S4	INT A12S7	INT A13S1	INT A14S5	INT A15S10
INT A11S5	INT A12S8	INT A13S2	INT A14S6	INT A15S11
...	...	...	...	INT A15S12
INT A11S10	INT A12S13	INT A13S4	INT A15S9	

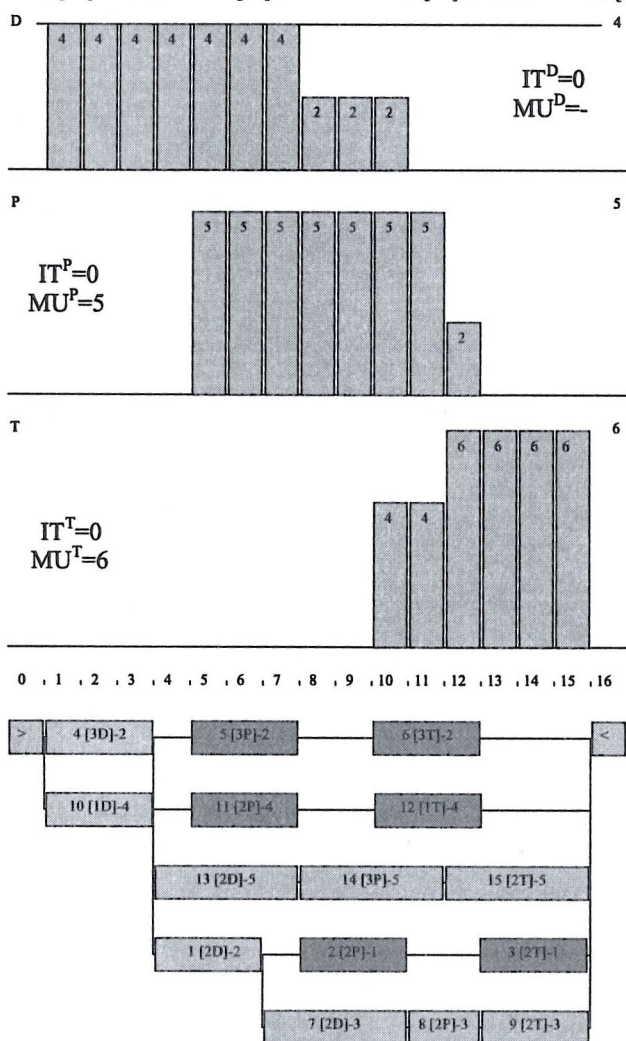


Az  $IT_{SH}$  modellben felírt  $SS$  feladat megoldására a *Lindo* rendszert használtuk, amely egy P133Mhz számítógépen 15 perc alatt végzett az 506 583 iterációval. A statisztikákból kitűnik, hogy a feladathoz tartozó mátrix sűrűsége, nagy méretének ellenére igen kicsi.

ROWS = 622    VARS = 416    INTEGER VARS = 135 (135 = 0/1)    NONZEROS = 2070  
 QCP = 0    DENSITY = 0.008    CONSTRAINT NONZ=1897 (1542=+-1)    MIN&MAX ABSVALUE= 1 & 12  
 OBJ = MIN    NUMBER OF <: 177    NUMBER OF =: 105    NUMBER OF >: 339

A kapott ideális ütemezés IT+MU célfüggvény szerinti minimuma 11. A hozzá tartozó ütemezés és erőforrás-kihasználtsági diagram az 5.10 ábrán látható:

1 [2D]-1  $\Rightarrow$  4    4 [3D]-2  $\Rightarrow$  1    7 [2D]-3  $\Rightarrow$  7    10 [1D]-4  $\Rightarrow$  1    13 [2D]-5  $\Rightarrow$  4  
 2 [2P]-1  $\Rightarrow$  8    5 [3P]-2  $\Rightarrow$  5    8 [2P]-3  $\Rightarrow$  11    11 [2P]-4  $\Rightarrow$  5    14 [3P]-5  $\Rightarrow$  8  
 3 [2T]-1  $\Rightarrow$  13    6 [3T]-2  $\Rightarrow$  10    9 [2T]-3  $\Rightarrow$  13    12 [1T]-4  $\Rightarrow$  10    15 [2T]-5  $\Rightarrow$  12



5.10 ábra: Az  $SS$  feladat MILP modelljének MU+IT szerinti optimális megoldása



## ERŐFORRÁS KIEGYENLÍTÉS IMPLICIT LESZÁMLÁLÁSSAL

Az előző fejezetben bemutatott  $IT_{SH}$  modell és annak MILP módszerrel történő megoldása – mint az a futási eredményekből is látszik – csak kis méretű projektek esetén kivitelezhető. Éppen ezért szükséges egy olyan modellt és optimalizálási módszert is kidolgozni, amely hatékony megoldási eszköze lehet a többfajta erőforrást is igénylő, *közepes* méretű ütemezési feladatoknak. Ez a fejezet is az előző fejezet logikai felépítését követi, amennyiben itt is előbb egy alapmodellt ismertetünk, amely még nem képes a gyenge és erős erőforrások kezelésére, majd megfelelő módon kiterjesztjük azt. Először tehát megadunk az erőforrás kiegyenlítési problémák körére egy általános, az újfajta állásidőkön (IT) alapuló modellt, a hozzá tartozó implicit leszámlálási algoritmussal együtt. Majd ezt továbbfejlesztjük egy olyan kétfázisú implicit leszámlálási algoritmussá, amely képes kezelni a gyenge és erős erőforrások esetét is. Az alapalgoritmust egy hatékony vágási szabályokkal ellátott keresőfa módszer formájában írjuk le, míg a második esetben egy kétszintű, keresőfák-keresőfája struktúra lesz a célravezető.

## 6.1 Implicit leszámlálási alapmodell

Jelölje most is  $\mathfrak{R}$  annak a  $k$ -féle erőforrásnak a halmazát, amelyekre a projekt(ek) kivitelezése során szükségünk lesz. Az alapmodellben feltételezzük, hogy ezen erőforrások mindegyike megújuló erőforrás abban az értelemben, hogy mindegyik  $r \in \{1, 2, \dots, k\}$  erőforrásból végtelen sok erőforrásegység áll a rendelkezésünkre. Valamint itt is alkalmazzuk a 4. fejezetben bevezetett üres nyitó és záró tevékenységet.

Fontos megjegyezni, hogy az egy erőforrás esetén alkalmazott a „kvázikonkáv profilt legjobban megközelítő” megoldás fogalma több erőforrás esetén nem egyértelmű, és többféleképpen is értelmezhető. A mi esetünkben ez azt jelenti, hogy „a megoldás” helyét a „hatékony megoldások” veszik át. Másképpen szólva a probléma Pareto optimumait keressük. Tehát célunk az 5.2 fejezetben bevezetett állásidő mérték szerinti összes hatékony megoldás felkutatása.

### Definíció

Az  $S \in \mathcal{S}$  ütemezést *hatékony ütemezésnek* nevezzük, ha nem létezik olyan  $S' \in \mathcal{S}$  ütemezés, amelyre minden  $IT^r(S') \leq IT^r(S)$  egyenlőtlenség teljesül úgy, hogy az  $r = 1, 2, \dots, k$  esetek legalább egyikében az egyik egyenlőtlenség szigorú.

Az eddigiek ismeretében az erőforrás-kihasználtság optimalizálási probléma modellje a következő alakban adható meg:

A projekt határidejének ( $T$  időtartamának) rögzítése mellett,  
a lehetséges ütemezések terében keressük azokat az  $S \in \mathcal{S}$  elemeket, amelyek  
az  $IT(S) = (IT^1(S), IT^2(S), \dots, IT^k(S))$  szerint hatékony megoldások.

## 6.2 Leszámlálási algoritmusok és keresőfák

Az ütemezési feladatok diszkrét tulajdonsága lehetőséget ad arra, hogy a valamilyen szempontból kívánatos ütemezéseket úgy keressük meg, hogy sorra megvizsgáljuk az összes elképzelhető ütemtervet. Az összes ütemezés explicit felsorolása természetesen a feladatok méretéből fakadóan, belátható időn belül lehetetlen. A kezelhetőség érdekében két dolgot kell megoldani: (1) az ütemezéseket bizonyos ismérvek alapján csoportosítani kell, hogy a megvizsgálendő esetek számát jelentősen csökkenteni tudjuk; valamint (2) olyan adatstruktúrát kell találnunk, amely lehetőséget ad ezen dinamikus csoportok ha-

tékony nyilvántartására. Ennek programozás-technikailag az egyik leghatékonyabb megoldása a fa-struktúra alkalmazása.

A keresőfák olyan adatszerkezetek, amelyekre a dinamikus halmazok számos műveletét értelmezzük, többek között a KERES, MINIMUM, MAXIMUM, ELŐZŐ, KÖVETKEZŐ, BESZÚR és TÖRÖL műveleteket. Ezért a keresőfák mind szótárként, mind elsőbbségi sorként használhatóak [6.5]. A modell segítségével megfogalmazott feladatok szempontjából a keresőfák alkalmazásának egyik legnagyobb haszna az, hogy szükség esetén számos, a szakirodalomban ismert algoritmus (vágási szabály) és heurisztika implementálható.

Az ütemezések osztályozása is természetes módon oldható meg. Közös osztályba sorolhatóak azok az ütemezések, amelyeknél bizonyos tevékenységek ugyanazon időpontban kezdődnek; vagy egy másik lehetőség szerint azok, melyek ugyanazon közvetlen megelőzési relációknak tesznek eleget. Egy-egy ilyen halmaz reprezentánsaként mindig azt az ütemezést tüntetjük fel, amelyben a halmaz által nem rögzített tevékenységek ütemezése a lehető legkorábbi (ES). A keresőfa csúcspontjai ennek megfelelően ilyen halmazokat jelölnek. A fa egy csúcsának *kibontása* alatt azt értjük, hogy onnan olyan leágazásokat definiálunk, melyek a csúcspont részhalmazainak felelnek meg. A teljesen kibontott fa levelei már az egyes konkrét megoldásoknak felelnek meg; és fa bejárása valósítja meg a leszámlálást.

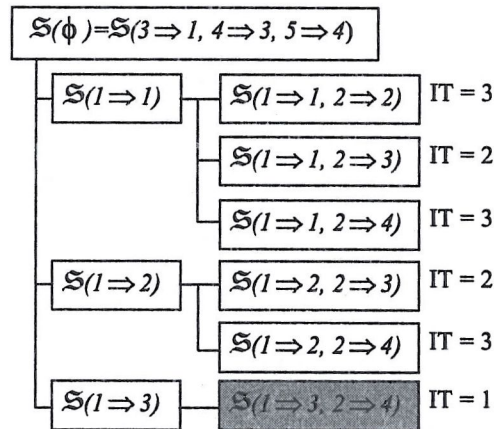
A 6.1 modell hatékony megoldásait előállító módszert tehát egy keresőfa formájában adjuk meg, ahol a fa csomópontjait egy-egy olyan halmazzal jellemezzük, amelyben néhány tevékenység kezdete explicit módon rögzítve van. Azt mondjuk, hogy egy tevékenység „*mozgatható*”, ha egyrészt nem kritikus tevékenység, másrészt a kezdete nincs explicit módon lerögzítve [6.1].

A keresőfa csúcsaihoz rendelt, a kezdési időpontokat rögzítő halmazokat  $F$ -fel jelöljük. Az  $F$  halmaznak és a közvetlen megelőzési relációk halmazának ( $E$ ) együttesen eleget tevő lehetséges ütemezések halmazát  $S(F)$  jelöli. Az  $S(F)$



ütemezés-halmaz reprezentáns elemének jele  $ES(\mathcal{S}(F))$ , amelyben a mozgatható tevékenységek kezdete a CPM szerinti legkorábbi ütemezés. A továbbiakban ezekre az ütemezés-halmazokra (az  $F$  halmaz ismeretében) mindig a reprezentánsaikkal hivatkozunk. Ezzel összhangban a megoldások száma alatt mindig a reprezentáns megoldások számát értjük, és ennek megfelelően az ábrákon is csak ezeket az ütemezéseket tüntetjük fel.

A keresőfa gyökeréhez tartozó halmaznak az  $F = \emptyset$  üres halmazt választva a lehetséges ütemezések halmazát kapjuk:  $\mathcal{S} = \mathcal{S}(\emptyset)$ , melyet a hagyományos, CPM szerinti legkorábbi ütemezés reprezentál. Ha ez az ütemezés éppen a határidőre készül el, akkor a kritikus útvonal elemeit a megfelelő  $F$ -halmaz megadásával akár rögzíthetjük is. Ebben a csomópontban még minden, eddig nem rögzített tevékenység mozgatható.



6.1 ábra: Az 5.8 ábrán bemutatott kisméretű feladatoz tartozó keresőfa és az optimális megoldáshoz tartozó levél

Egy olyan csomópont kibontása, amelyben vannak mozgatható tevékenységek, lényegében az adott csomópont annyiféle alábontását jelenti, mint ahányféleképp a kiszemelt mozgatható tevékenység kezdetét rögzíteni lehet. A keresőfa levelei ennek megfelelően olyan ütemezéseknek felelnek meg, ahol már vagy nincs mozgatható aktivitás, vagy a később ismerttetendő „legalább olyan jó” vágási szabály miatt a további leágazások tiltottak. Az algoritmus determinisztikusságának megőrzése érdekében a kibontást mindig a topológiai értelemben

vett első (legalacsonyabb sorszámú mozgatható) aktivitás szerint végezzük. Ez a sorrend lényegében a fa szélességi illetve mélységi bejárásának felel meg [6.5], de alkalmazható más bejárési stratégia is. Ehhez mindössze a fa csúcspontjait kell megfelelően súlyozni, illetve sorrendezni. Ilyenkor a keresőfa kibontása során a hatékony ütemezések halmaza folyamatosan változik.

A következőkben egy ilyen súlyozási eljárást mutatunk be. Ha az  $S \in \mathcal{S}$  ütemezés súlyfüggvényének az alábbi  $QC$  kvázi költséget választjuk,

$$QC(S) = \sum_{r=1}^k IT^r(S) = \sum_{r=1}^k \sum_{t=1}^T (\bar{U}_t^r(S) - U_t^r(S))$$

akkor a bejárást érdemes mindig a legalacsonyabb költségű csomópont kibontásával folytatni, hiszen ezek állnak legközelebb a minimumhoz. Ha az  $i$ . csomóponthoz több ütemezés is tartozik, akkor a  $QC(\mathcal{S}^*) = QC(S^*)$  költség alkalmazható, ahol  $S^*$  egy tetszőlegesen választott  $S^* \in \mathcal{S}^*$  ütemezés. A determinisztikusság megőrzése érdekében előírhatjuk az  $S^* = ES(\mathcal{S}^*)$  reprezentáns kiválasztását.

Természetesen az állásidők összegén kívül bármely olyan költségfüggvény alkalmazható, amely a projektmenedzser vagy a döntéshozó preferenciáit jobban kifejezi. A megfelelő költségfüggvény kialakítása után a keresőfa kibontását mindig annál, a még ki nem bontott csomópontnál tudjuk folytatni, amely a döntéshozó szempontjából a legígéretesebb.

### 6.3 Alsó becslések és vágási szabályok

Egy adott  $F$  halmaz mellett legyen  $C(F) \subset \{1, 2, \dots, T\}$  a projekt időegységeinek egy olyan halmaza, amely időpontokban az erőforrás-kihasználtsági profil a mozgatható tevékenységek segítségével nem változtatható meg. Formálisan azt mondhatjuk, hogy  $t \in C(F)$  pontosan akkor teljesül, ha az  $U_t^r(S)$ ,  $r \in \{1, 2, \dots, k\}$  értékek függetlenek az  $S \in \mathcal{S}(F)$  ütemezés megválasztásától [6.2].

Azon időpontokban, ahol ez az érték nem konstans, azaz  $t \notin C(F)$ , ott az erőforrásonként előforduló minimális  $U'_t(S)$  értéket az  $r$ . erőforrás felhasználásának  $t$ . időpontbeli *konstans részének* nevezzük. A konstans rész definíciója a következőképpen adható meg:

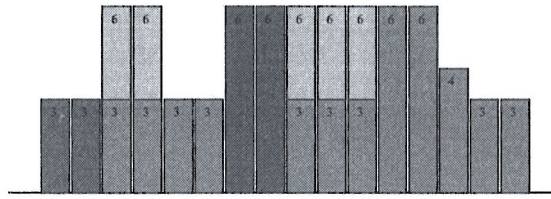
$$C'_t(F) = \min(U'_t(S) : S \in \mathcal{S}(F)) \text{ ahol } t \in \{1, 2, \dots, T\} \text{ és } r \in \{1, 2, \dots, k\}$$

Nyilvánvaló, hogy minden rögzített  $r \in \{1, 2, \dots, k\}$ ,  $t \in \{1, 2, \dots, T\}$  értékpár mellett, és tetszőleges  $S \in \mathcal{S}(F)$  ütemezés esetén igaz a  $0 \leq C'_t(F) \leq U'_t(S)$  egyenlőtlenség. Továbbá  $t \in C(F)$  akkor és csak akkor, ha a  $C'_t(F) = U'_t(S)$  egyenlőség minden  $r \in \{1, 2, \dots, k\}$  és  $S \in \mathcal{S}(F)$  esetén teljesül.

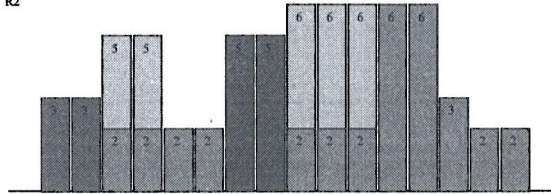
Illusztrációként tekintsük újra az 5.1 fejezetben már ismertetett *ESS* feladatot, és vizsgáljuk meg, hogyan szemléltethetőek a fenti fogalmak az  $F = \{6 \Rightarrow 7\}$  halmaz esetén, mely szerint a 6. tevékenységet a 7. időpontban rögzítjük. A feladatban az 1-3-4-5-7-10 tevékenységek jelölik ki a kritikus utat, így azok akár rögzíthetőek is. Az  $F$  halmaz által definiált  $S_6 = 7$  rögzítés után csak a 2., a 8. és a 9. tevékenységek mozgathatóak. Az erőforrás-kihasználtsági diagramon a  $C(6 \Rightarrow 7) = \{1, 2, 7, 8\}$  halmazhoz tartozó oszlopokat sötétszürkével jelöltük, és könnyen látható, hogy ezek tényleg invariánsak a mozgatható tevékenységek „tologatására”. A többi oszlop konstans  $C'_t(6 \Rightarrow 7)$  része szürke, amely természetesen más és más lehet a két erőforrás esetében. Az erőforrás-kihasználtsági diagram oszlopainak fennmaradó részét pedig világosszürkével jelöltük.



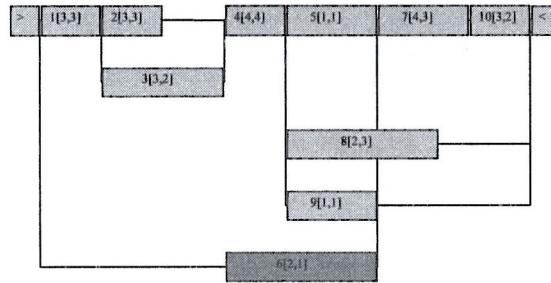
R1



R2



0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17



6.2 ábra: Ütemezéstől független, konstans erőforrás-kihasználtsági részek

### 6.3.1 A „legalább olyan jó” vágási szabály

A következőkben megmutatjuk, hogy a kvázi-konkávitás rendelkezik egy speciális monotonitási tulajdonsággal, amely lehetővé teszi a címben említett vágási szabály bevezetését.

Legyen  $\mathcal{S}(F) \subseteq \mathcal{S}$  egy tetszőleges ütemezési halmaz, amelyen vezessük be minden  $r \in \{1, 2, \dots, k\}$  erőforrás esetén a következő jelöléseket:

$$MAXCL_t^r = \max(C_1^r(F), C_2^r(F), \dots, C_t^r(F)), \text{ ahol } t \in \{1, 2, \dots, T-2\}, \text{ és}$$

$$MAXCR_t^r = \max(C_t^r(F), C_{t+1}^r(F), \dots, C_T^r(F)), \text{ ahol } t \in \{3, 4, \dots, T\}.$$

### 6.1 Lemma

Egy adott  $\mathcal{S}(F)$  ütemezési halmaz és  $r \in \{1, 2, \dots, k\}$  esetén az alábbi képlettel definiált  $LIT^r(F)$  értékek

$$LIT^r(F) = \sum_{t \in C(F)} \max(\min(MAXCL_{t-1}^r, MAXCR_{t+1}^r) - U_t^r, 0)$$

az állásidők alsó becslését adják, azaz minden  $S \in \mathcal{S}(F)$  ütemezésre  $LIT^r(F) \leq IT^r(S)$ .

**Bizonyítás:** A bizonyítás az  $LIT^r(F)$  valamint az  $IT^r(S)$  értékek definíciójának összevetéséből közvetlenül adódik, hiszen egyrészt éppen a konstans rész definiálásakor állapítottuk meg, hogy a  $0 \leq C_t^r(F) \leq U_t^r(S)$  egyenlőtlenség mindig teljesül, vagyis  $MAXCL_t^r \leq MAXL_t^r$ ; másrészt a  $LIT^r(F)$  definíciójában szereplő összegben kevesebb pozitív összeadandó szerepel, mint az  $IT^r(S)$  képletében.

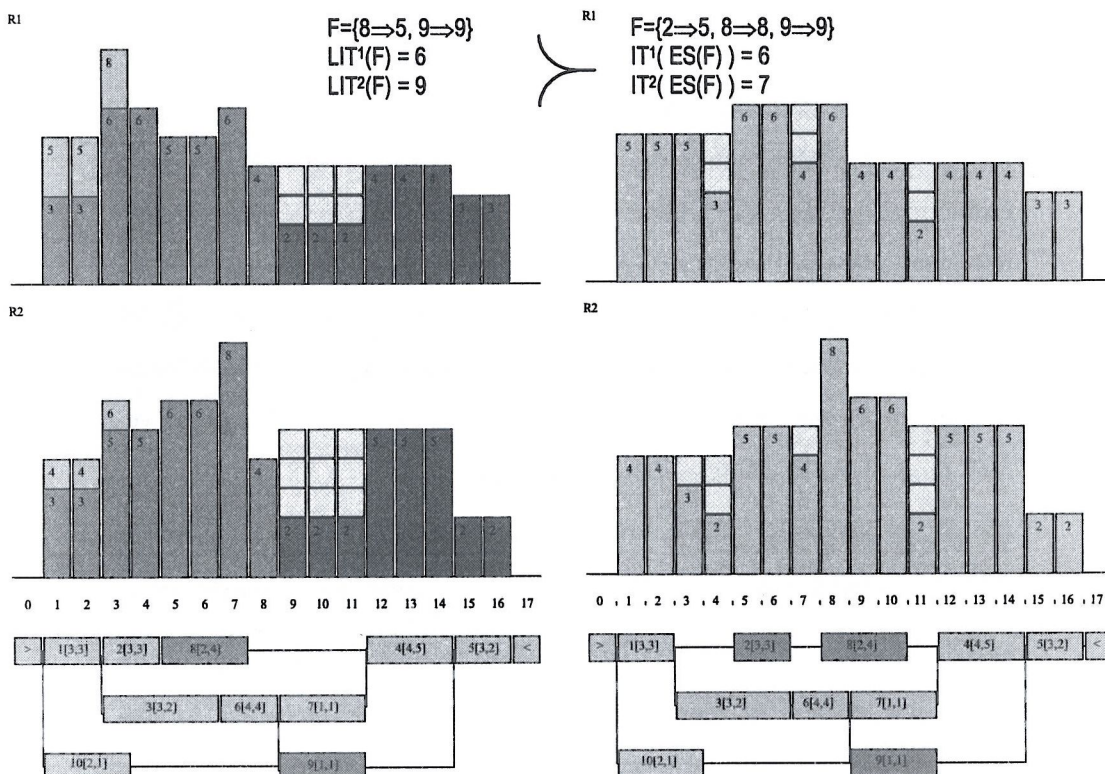
$$\begin{aligned} IT^r(S) &= \sum_{t=1}^T (\bar{U}_t^r(S) - U_t^r(S)) = \sum_{t=1}^T \max(\min(MAXL_{t-1}^r, MAXR_{t+1}^r) - U_t^r, 0) \geq \\ &\geq \sum_{t=1}^T \max(\min(MAXCL_{t-1}^r, MAXCR_{t+1}^r) - U_t^r, 0) \geq \\ &\geq \sum_{t \in C(F)} \max(\min(MAXCL_{t-1}^r, MAXCR_{t+1}^r) - U_t^r, 0) = LIT^r(F) \quad \diamond \end{aligned}$$

A lemmában definiált  $LIT^r(F)$  kifejezés szemléletes jelentése a következő: Vegyük az  $r$ . erőforrás-kihasználtsági diagramot, és töröljük belőle a nem konstans (világosszürke) részeket. Az így kapott csonka diagram és annak kvázi-konkáv burka közötti területek közül csak azokat vegyük figyelembe, amelyek a fix (sötétszürke) részek felett helyezkednek el. Ezek az erőforrás felhasználási állásidők a mozgatható tevékenységek segítségével már biztosan nem változtathatóak meg, ezért lesz a  $LIT^r(F)$  alkalmas betölteni az alsó korlát szerepét (Lower bound for Idle Times)

A bizonyításból az is könnyen leolvasható, hogy ha az  $F$  halmazban már minden tevékenység kezdete rögzített, azaz  $\mathcal{S}(F) = \{S\}$  egyelemű halmaz, akkor

$LIT^r(F) = IT^r(S)$ . Így ha a keresőfa csomópontjaihoz rendelt halmazok kiértékelő függvényeként a  $(LIT^1(F), LIT^2(F), \dots, LIT^k(F)) = LIT(F)$  vektort választjuk, akkor a fa leveleinél kapott értékek éppen az adott ütemezés erőforrásonként állásidejét adják.

A 6.1. lemma segítségével a „legalább olyan jó” vágási szabályt a következőképp határozhatjuk meg: Az  $S^*$  ütemezés-halmaz legalább olyan jó, mint az  $S(F)$  - röviden  $S^* \prec S(F)$  -, ha minden  $r \in \{1, 2, \dots, k\}$  erőforrásra nézve teljesül az  $IT^r(S^*) \leq LIT^r(F)$  egyenlőtlenség, ahol  $S^* = ES(S^*)$  jelöli az  $S^*$  ütemezés-halmaz reprezentáns elemét. Ilyen esetben ha  $S^*$  már szerepel csomópontként a keresőfában, akkor az  $S(F)$ -hez tartozó csomópont kibontása elhagyható.



6.3 ábra: Példa a „legalább olyan jó” szabály alkalmazására, ahol a jobb oldal dominálja a bal oldalt.



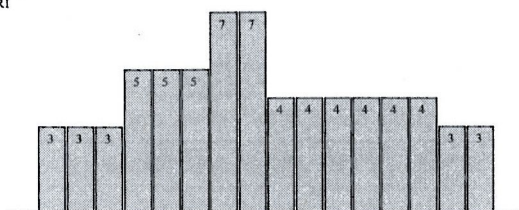
A „*legalább olyan jó*” szabály alkalmazását a 6.3 ábra szemlélteti. Az ábrán feltüntetett ütemezéseknél a sötétebb színű tevékenységek rögzítettek, ennek következtében mindkettőnél csak a 10-es tevékenység mozgatható. Az egy-ségnégyzetekről könnyen leolvasható, hogy a jobb oldali *ES* ütemezés legalább olyan jó, mint bal oldali ütemezési halmaz.

Ez a „*Csébfalvi*” vágás rendelkezik azzal az előnyös tulajdonsággal is, hogy a keresés folyamán, az egyre korlátozottabb ütemezési halmazokon egyre hatékonyabban alkalmazható. [6.4]

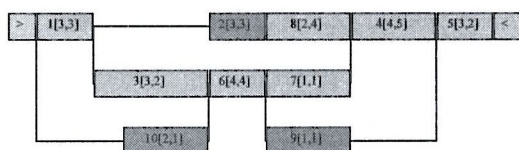
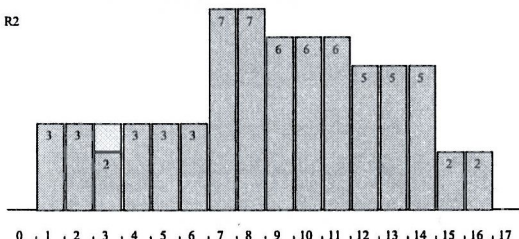
A 6.4 ábrán nyomon követhetjük az algoritmus működését, ahol demonstrációs és ábrázolhatósági okokból nem az eredeti *ESS* feladat, hanem annak egy kissé módosított változatához tartozó keresési illetve relaxált keresési fa (lásd 6.3.2 fejezet) látható.

Az ábra bal oldalán látható, kiterjesztett Gantt diagrammal megadott feladathoz tartozó relaxálás nélküli fa az ábra jobb oldalán található. A keresőfa minden csomópontjához annak egy tömör jellemzését is megadtuk. Legelől a csomópont sorszáma van feltüntetve, amely alapján a fa felépítésének története rekonstruálható. Például a 0. gyökérpont kibontása után keletkeztek az 1., ..., 5. csomópontok. A sorszám után nem magát az  $F$  halmazt tüntettük fel, hanem annak mindig csak a legújabb elemét. Ennek megfelelően az 57. csomópont-hoz tartozó  $F$  halmazt úgy kaphatjuk meg, ha a gyökértől a hozzá vezető úton szereplő elemeket összesítjük, azaz  $F_{57} = \{2 \Rightarrow 3, 8 \Rightarrow 7, 10 \Rightarrow 2\}$ . Az 57. csomópont adatait tovább vizsgálva azt olvashatjuk le, hogy  $LIT(F_{57}) = (6, 8)$ , valamint az  $\mathcal{S}(F_{57})$  halmaz legkorábbi *ES* ütemezésénél az állásidők értékei  $IT(ES(F_{57})) = (10, 14)$ . Mivel azonban az 57. csúcspont egyben levél is, ezért az *ES* reprezentáns ütemezés maga „az” ütemezés. Az állapotjelző valójában technikai segédeszköz, amiből az 57. csúcspont esetében megtudhatjuk, hogy ezt egy alacsonyabb sorszámú (még hozzá pontosan a 4.) csúcspont dominálta, azaz metszette le.

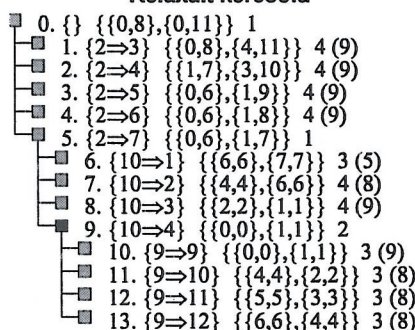
R1



R2



### Relaxált keresőfa



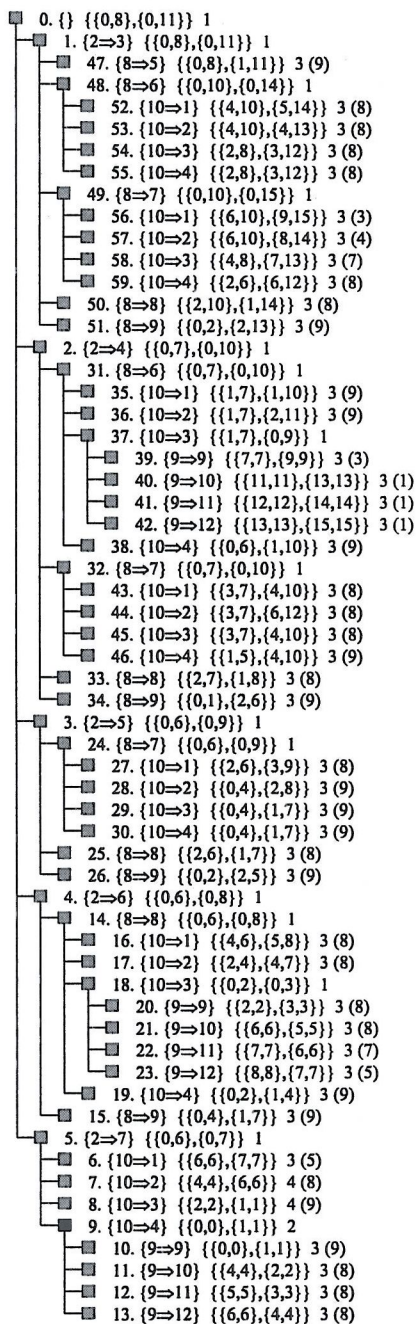
A relaxált keresőfában a LIT<sub>r</sub> értékeket a relaxált IT alapmodell alsó becsléseivel helyettesítettük.

### Jelmagyarázat

Csomópont. {F halmaz bővítés} { {LIT<sub>1</sub>, IT<sub>1</sub>}, {LIT<sub>2</sub>, IT<sub>2</sub>} }  
Állapotjelző (Melyik ág metszette le)

### Az állapotjelző értékei

- |                      |                         |
|----------------------|-------------------------|
| 0 Generált           | 2 Hatékony              |
| 1 Megvizsgált        | 3 Előző által lementett |
| 4 Új által lementett |                         |



6.4 ábra: Egy mintafeladat IT optimális ütemezése, illetve a relaxált és relaxálás nélküli keresőfák

Lássuk hogyan:

$$57. \{LIT^1(F_{57}) = 6, IT^1(ES(F_{57})) = 10\}, \{LIT^2(F_{57}) = 8, IT^2(ES(F_{57})) = 14\}$$

$$4. \{LIT^1(F_4) = 0, IT^1(ES(F_4)) = 6\}, \{LIT^2(F_4) = 0, IT^2(ES(F_4)) = 8\}$$

Amiből kapjuk, hogy egyszerre teljesül a  $6 = IT^1(ES(F_4)) \leq LIT^1(F_{57}) = 6$  és a  $8 = IT^2(ES(F_4)) \leq LIT^2(F_{57}) = 8$  egyenlőtlenség, vagyis  $\mathcal{S}(F_4) \prec \mathcal{S}(F_{57})$ .

Már csak annak az illusztrálása maradt hátra, hogy az aktuális csomópontok közül miként választottuk ki a soron következő kibontandót. Ehhez csúcsonként kiszámítottuk az  $IT(F) = IT^1(ES(F)) + IT^2(ES(F))$  összegeket, és ahol ez a legkisebb volt (holtverseny esetén a legkisebb sorszámút) bontottuk tovább. Mivel  $IT(F_1) = 19$ ,  $IT(F_2) = 17$ ,  $IT(F_3) = 15$ ,  $IT(F_4) = 14$  és  $IT(F_5) = 13$ , ezért az 5. csomópont kibontása következett.

### 6.3.2 Az „LP relaxálás” vágási szabály

A lineáris programozási (LP) feladatok olyan problémák, melyek polinomiális idő alatt megoldhatóak, míg a 0/1-értékű (0/1-LP), illetve az egész értékű programozási feladatok (IP) NP nehézségű problémák. Ez különösen akkor érdekes, ha a feladatok  $A \cdot X = b$  feltételrendszerében ugyanaz az  $A$  mátrix és  $b$  vektor szerepel, továbbá a  $X \cdot c^T$  célfüggvényünkben is azonos a  $c$  vektor. Az egyetlen különbség ilyenkor mindössze annyi, hogy egy LP probléma esetén a megoldásokat a valós számok, 0/1-LP probléma esetén a  $\{0,1\}$  halmazon, míg IP feladatok esetén az egész számok halmazán keressük. Ez a különbség a leglényegesebb elem, hiszen ez határozza meg a feladat megoldásának nehézségi fokát. Ez a magyarázata annak, hogy miért olyan népszerű a lineáris programozás eszköze a kombinatorikus optimalizálási feladatok körében.

Mivel az LP feladatok polinomiális idő alatt megoldhatóak, míg a 0/1-LP és IP feladatok NP nehézségűek, így teljesen természetes ötlet az, hogy a 0/1-LP illetve IP feladatokat LP feladatként oldjuk meg. Ezt a technikát *relaxációnak*



nevezzük, hiszen a  $\{0,1\}$  halmazon, illetve az egészek halmazán keresett optimális megoldásokat immár a teljes valós számok körében keressük. Természetesen az így kapott LP megoldások nem feltétlenül lehetséges megoldásai az eredeti feladatnak, de minimum célfüggvény esetén alkalmas alsó (maximum feladatok esetén felső) becslésekhez jutunk. [6.6]

A relaxáció módszerét itt is alkalmazhatjuk, hiszen az előző fejezetben az erőforrás kiegyenlítő problémákra már megadtuk az  $IT$  illetve  $IT_{SH}$  (MILP) modelleket.

## 6.2 Lemma

Az  $\mathcal{S}(F)$  csomóponthoz tartozó „*legalább olyan jó*” vágási szabálynál alkalmazott  $LIT$  alsó becslés vektort helyettesíthetjük azon erőforrás kiegyenlítési feladat  $IT$  *alapmodell* szerinti relaxált megoldásának célfüggvény értékeivel, amelyet úgy kapunk az eredeti feladat  $F$ -beli tevékenységeit az ott megadott időpontokban rögzítjük. Relaxáláskor mindössze az  $LLTR'_i$  és  $AS_{i,t}$  változókra megadott 0/1 kritériumokat kell helyettesíteni a  $0 \leq LLTR'_i \leq 1$  és  $0 \leq AS_{i,t} \leq 1$  feltételekkel.

**Bizonyítás:** Az  $IT$  alapmodell feltételrendszerének konstrukciója és célfüggvénye pontosan megfelel a vágási szabály előírásainak. ♦

A keresés során érdemes tehát az  $IT$  alapmodell relaxált változatát használni, hiszen a becslések elkészítésének gyorsasága kritikus tényező. Ha most visszatekintünk a 6.4 ábra bal alsó részére, akkor megtaláljuk a feladathoz tartozó azon keresőfát, amelynél a  $LIT$  értékek helyett a relaxált feladatok adták az alsó becsléseket. Már az 1. csomópont esetében is 0-ról egyből 4-re tudtuk emelni az alsó becslést a második erőforrás esetében. A relaxált fa méretén jól megfigyelhető a vágási szabály hatékonysága. A vágási szabály hatékonyságának javításával a keresőfa az eredeti 60 csomópont helyett - a relaxálással kapott alsó becslések következtében - mindössze 14 csomópontra zsugorodott össze.

## 6.4 Implicit leszámhlási modell gyenge és erős erőforrások esetén

Az előbb ismertetett implicit leszámhlásos módszer kiterjesztése gyenge és erős erőforrások esetére már nem olyan egyértelmű, mint az a vegyes egészértékű modell esetében volt. A most ismertetendő több kritériumos optimalizálási modell és a megoldására javasolt kétfázisú implicit leszámhlási algoritmus külön érdekessége, hogy sikeresen ötvözi az ütemezési feladatok két nagy problémakörét: a hozzárendelési (RAP – Resource Allocation Problem) és a kiegyenlítési (RLP – Resource Leveling Problem) feladatok jellegzetességeit.

Ebben a modellben is ugyanolyan szerkezetű (gyenge és erős) erőforrásokkal, valamint ugyanazzal a kétféle globális célfüggvénnyel (minimális állásidő és maximális kihasználtság) fogunk dolgozni, mint tettük azt az 5. fejezetben. Az eddig használt jelöléseknek megfelelően feltételezzük, hogy a modellben az erős erőforrások indextartománya  $1, 2, \dots, k$ , a gyenge erőforrásoké pedig  $k + 1, k + 2, \dots, k + m$ . Ha az erős erőforrások mindegyikére nézve elfogadható ütemezések halmazát  $\mathcal{S}^H$ -val jelöljük, akkor az erős és gyenge erőforrások állásidőjét és maximális kihasználtságát minimalizáló probléma a következő alakban adható meg:

Állapítsuk meg azt a minimális  $T$  időtartamot, amely mellett már létezik elfogadható ütemezés, majd az *elfogadható ütemezések* terében keressük meg az

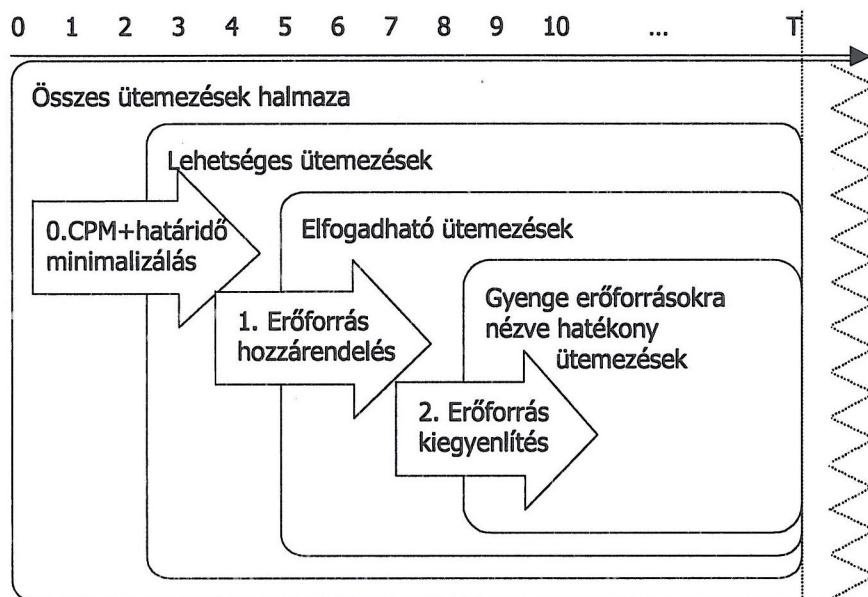
$$(IT^{\mathfrak{R}}(S), MU^{\mathfrak{R}}(S)) = (IT^1(S), \dots, IT^{k+m}(S), MU^1(S), \dots, MU^{k+m}(S))$$

vektor szerinti összes hatékony (Pareto optimális)  $S \in \mathcal{S}^H$  megoldást.

Elképzelhető a fenti modellnek egy olyan *gyengített* változata is, ahol a célvektorban csak a gyenge erőforrásokhoz tartozó  $(IT^{\mathfrak{R}_s}(S), MU^{\mathfrak{R}_s}(S))$  értékek szerepelnek, lévén az erős erőforrásokhoz tartozó  $(IT^{\mathfrak{R}_H}(S), MU^{\mathfrak{R}_H}(S))$  értékek szintje a modell természeténél fogva eleve korlátozott.

## 6.5 Kétfázisú implicit leszámítási algoritmus

A fenti modellben felírt problémák megoldására a 6.5 ábrán vázolt kétfázisú módszer alkalmazható, ahol az első fázis egy hozzárendelési (RAP), míg a második fázis több kiegyenlítési (RLP) probléma megoldását jelenti. Ez a kétfázisú módszer azonban már közepes méretű feladatok megoldására is alkalmas.



6.5 ábra: A kétfázisú, implicit leszámítási algoritmus vázlata

Az egyes fázisok végrehajtásához tetszőleges olyan módszer használható, amely képes a keresőfák-keresőfája struktúra kezelésére. Az általunk javasolt algoritmus mindkét fázisa a 6.2 szakaszban ismertetett eljárás egy-egy módosított változatára épül. Először a fa csomópontjaihoz rendelt halmazok megválasztását cseréljük le úgy, hogy képesek legyünk az elfogadható ütemezések előállítására; majd az algoritmust módosítjuk úgy, hogy alkalmas legyen többféle célkritérium szerinti optimumok megkeresésére is.

### 6.5.1 Első fázis – Hozzárendelési feladat

Az első fázisban egy CPM ütemezésből kiindulva az elfogadható ütemezések terét állítjuk elő. Az erős erőforrásokra nézve elfogadható ütemezések megke-



resése egy olyan hozzárendelési probléma megoldását jelenti, amelynél a gyenge erőforrásokat figyelmen kívül hagyjuk.

### Definíció

Ha egy  $S \in \mathcal{S}$  ütemezés egy  $t$  időpontban valamelyik  $r \in \{1, 2, \dots, k\}$  erős erőforrásból többet igényel, mint amennyi a rendelkezésre áll  $U'_t(S) > L' = R(t)$ , akkor ezt *erőforrás konfliktusnak* nevezzük.

### Definíció

Ha az  $S \in \mathcal{S}$  ütemezés egy  $t$  időpontjában egy  $r \in \{1, 2, \dots, k\}$  erős erőforrásra nézve erőforrás konfliktus adódik, akkor az adott időpontban aktív  $i \in Act_S(t)$  tevékenységek közül azoknak a halmazát, amelyek igénylik az  $r$  erőforrást, azaz  $R'_i > 0$  - *összeférhetetlenségi* halmaznak nevezzük.

### Definíció

Legyen  $K$  egy  $r \in \{1, 2, \dots, k\}$  erős erőforrás  $S \in \mathcal{S}$  ütemezéséhez tartozó összeférhetlenségi halmaza. Ekkor a  $K' \subseteq K$  részhalmazát *minimális összeférhetlenségi* halmaznak nevezzük, ha a  $K'$ -beli tevékenységek még összeférhetlenek, azaz  $\sum_{i \in K'} R'_i > L'$ ; de bármely valódi  $K'' \subset K'$  részhalmaza már összeférhető, vagyis  $\sum_{i \in K''} R'_i \leq L'$ .

### Definíció

Legyen  $G = (V, E)$  egy AoN gráf, valamint  $K$  egy minimális összeférhetlenségi halmaz. Ekkor az  $E' = \bigcup \{i \rightarrow j : i, j \in K\}$  alakú közvetlen megelőzési relációkból álló halmazokat *konfliktus feloldó* halmazoknak nevezzük, a  $G' = (V, E \cup E')$  gráf lehetséges ütemezéseinek halmazát pedig  $\mathcal{S}[E']$ -vel jelöljük.

Nyilvánvaló, hogy  $S[\emptyset] = S$ ,  $S[E'] \subseteq S$ ; továbbá ha  $S[E'] = \emptyset$ , akkor az eredeti feladat  $E'$  halmazzal kiegészített változatának nem létezik elfogadható ütemezése. Az  $S[E']$  halmazt (az 5. fejezethez hasonlóan) most is a legkorábbi ütemezés szerinti ES reprezentánssal adhatjuk meg.

Ha a rendelkezésre álló erős erőforrások valamelyikének túlzott korlátozottsága miatt a kritikus út módszerével kapott legkorábbi ütemterv nem valósítható meg (mert az adott határidőn belül nem létezik elfogadható megoldás), akkor az adott erőforrások felhasználási konfliktusainak feloldásával (vagyis bizonyos tevékenységek ütemezésének késleltetésével) olyan ütemtervet kell keresnünk, amely egyrészt (1) az összes erős erőforráskorlátot kielégíti, másrészt (2) a kritikus út módszerével adódó globális befejezési időpontot minimális mértékben növeli. Ezt a lépést 0. fázisnak nevezzük, és azért nem térünk ki rá külön, mert e kettős kritériumnak megfelelő ütemezés ugyanazzal a technikával állítható elő, mint az erős erőforrásokra nézve elfogadható ütemezések halmaza (lásd az algoritmus 4. lépését).

Az első fázisban alkalmazott keresőfa csúcspontjaihoz most is halmazokat rendelünk hozzá, de ezek most nem tevékenységkezdetek rögzítési idejei lesznek, hanem újabb közvetlen megelőzési relációkat tartalmazó konfliktus feloldási halmazok.

### A keresési fa felépítése

Az itt ismertetendő fafelépítő algoritmus annyiban tér el a hagyományos algoritmusoktól, hogy alkalmas az összes elfogadható megoldás (ütemezés-osztály) megkeresésére. Leglényegesebb tulajdonsága pedig az, hogy megoldásnak csak azt az ütemezést tekint, amelyben minden lehetséges erőforrás konfliktus fel van oldva. Ez többet jelent annál, minthogy egy osztály minden eleme konfliktusmentes. Az algoritmus azt is biztosítja, hogy az általa meghatározott faleveleken, a közvetlen megelőzési relációknak megfelelő ütemezés-osztályokon belül, a tevékenységek kezdőpontjainak változtatásával erőforrás

konfliktus már nem következhet be. Ebből az a fontos tény következik, hogy az algoritmus által szolgáltatott elfogadható megoldások minden lehetséges késleltetése (mozgatható tevékenységeinek átütemezése) is elfogadható megoldás lesz. [6.3].

A keresési fa felépítésével kapcsolatos legfontosabb elemek az alábbiakban foglalhatóak össze:

1. A fa csúcspontjait az  $n = 0$  értékből kiindulva, azok keletkezési sorrendjében sorszámmal látjuk el. A fa gyökeréhez az üres konfliktus feloldási szabályhalmazt rendeljük, amely az eredeti feladatnak felel meg. A kiinduló állapothoz tartozó reprezentáns ütemezés ebben az esetben az erőforrás korlátok figyelembevétel nélkül kapott CPM szerinti legkorábbi ütemezés lesz.

2. Megállapítjuk az aktív csomópontokhoz tartozó első minimális összeférhetlenségi halmazokat. Az adott csomóponthoz rendelt konfliktus feloldó halmaznak megfelelő ütemezések közül némelyik továbbra is erőforrás konfliktus(oka)t okozhat. Az így kapott konfliktusok közül az „első” alatt mindig az időrendben legkorábbit értjük. Ekkor tudjuk, hogy a  $t$ . időpont előtt már nem fordulhat elő erőforrás konfliktus. Ha a  $t$ . időponthoz többféle minimális összeférhetlenségi halmaz is tartozik, akkor ezek közül tetszőlegesen jelölünk ki egyet. (Később ezen a véletlenszerű kijelölésen a heurisztikák hatékonyságának fokozása érdekében változtatni fogunk. A kiválasztási stratégiánkat az alábbi preferenciák alkalmazásával pontosíthatjuk: Válasszuk ki mindig (a) legkisebb elemszámú, (b) a legelőször megtalált, vagy (c) a legkönnyebben feloldható minimális összeférhetlenségi halmazt.)

3. A keresőfa aktív csomópontjai közül az egyik tetszőlegeset bontjuk ki. A kibontás ebben az esetben azt jelenti, hogy az adott csomópontból annyi leágazást készítünk, ahányféleképp a csomóponthoz tartozó minimális összeférhetlenségi halmaz elemei között páronként közvetlen megelőzési relációt definiálhatunk. Az újonnan keletkezett csomópontok konfliktus feloldási halmazai pedig a szülő ág feloldási halmazának az adott relációval kibővített



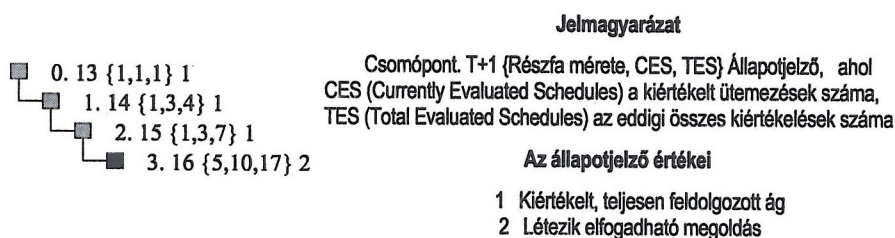
változatai lesznek. (A véletlenszerű elemkiválasztást itt is felválthatjuk a legígéretesebb csomópont kiválasztásával. Ilyenkor azt a csomópontot választhatjuk, amelyhez a 2. lépésben meghatározott minimális összeférhetetlenségi halmaz (a) a legkorábbi, (b) a legkisebb elemszámú, (c) amelynél a konfliktus halmazok száma a legkisebb volt, vagy egyszerűen (d) a legkisebb sorszámút. Ha az eljárás megkezdése elején listába rendezzük az összes lehetséges minimális összeférhetetlenségi halmazt, akkor maga a lista is használható prioritások kialakítására.)

4. A keresőfa kibontása akkor ér véget, ha minden levél vagy konfliktusmentes, vagy az adott határidő alatt feloldhatatlan konfliktust tartalmaz. A konfliktusmentes levelek adják az összes elfogadható ütemezések  $S^H$  osztályait. Ha az 1-3. lépések ismételtesével olyan fát kapunk, amelyben minden levél feloldhatatlan konfliktust tartalmaz, akkor tulajdonképpen a 0. fázisban vagyunk. Ekkor a fa gyökeréhez újabb gyökércsomópontot illesztünk, melyhez az eredeti feladatot rendeljük hozzá azzal a módosítással, hogy a  $T$  határidő értékét egységnivel megnöveljük.

A keresőfa kibontása során keletkező ágak száma dinamikusan növekszik. A kibontási lépéseknél felsorolt kiválasztási stratégiák azt a természetes feltevést tükrözik, hogy ha egy csúcspontban a lehetséges erőforrás konfliktusok száma kicsi, akkor ez a csúcspont *"valószínűleg közel van"* egy optimális megoldáshoz, amit a lehetséges konfliktusok teljes hiánya jellemez. A feltételezés természetesen nem garantálja egy adott kibontási lépés sikerességét, hiszen egészen kis lehetséges konfliktus szám esetében is felbukkanhatnak feloldhatatlan erőforrás konfliktusok, ami szükségszerűen visszalépéssel jár, de mindig olyan területekre irányítja az algoritmus figyelmét, ahol a sikeres befejezés valószínűsége a legnagyobb.

Természetesen az algoritmus első fázisa az első konfliktusmentes levél megtalálása után akár meg is állítható, mint ahogy azt a következő fejezet nagymértékű feladatának esetében kénytelenek is leszünk alkalmazni.

A továbbiakban a kétfázisú keresési fa algoritmust az 5.9 ábrán már ismertett *SS* mintafeladaton keresztül mutatjuk be. Ott ugyan nem említettük, de az 5.9 ábra egyben a keresőfa kiindulási CPM ütemezését mutatja. Mint a MILP modellben már utaltunk rá, a feladat az erős erőforrások túlzott korlátozottsága miatt legkevesebb csak 15 időegység alatt oldható meg. Ezért a megoldás a 0. fázissal kezdődik, melynek keresési fát a 6.6 ábrán láthatjuk. Az ábrán csak a gyökérpontok vannak feltüntetve. Az egyes gyökérpontokhoz tartozó ágak szerkezete minden esetben hasonló, ezért a 6.7 ábrán csak azt vizsgáljuk meg részletesebben, hogy az utolsó előtti, 2-es számú gyökérpont teljesen kibontott keresőfájából hogyan kapjuk meg a feketével jelölt 3-as számút.



6.6 ábra: Az *SS* feladat 0. fázisához tartozó tömörített keresési fa

Előtte azonban érdemes elkészíteni az összes lehetséges minimális összeférhetlenségi konfliktus halmaz kialakulásának 6.1-es táblázatát, amely egyben lexicografikus rendezést is definiál. A táblázat egyes sorainak olvasatát a második sorának értelmezésével mutatjuk be.

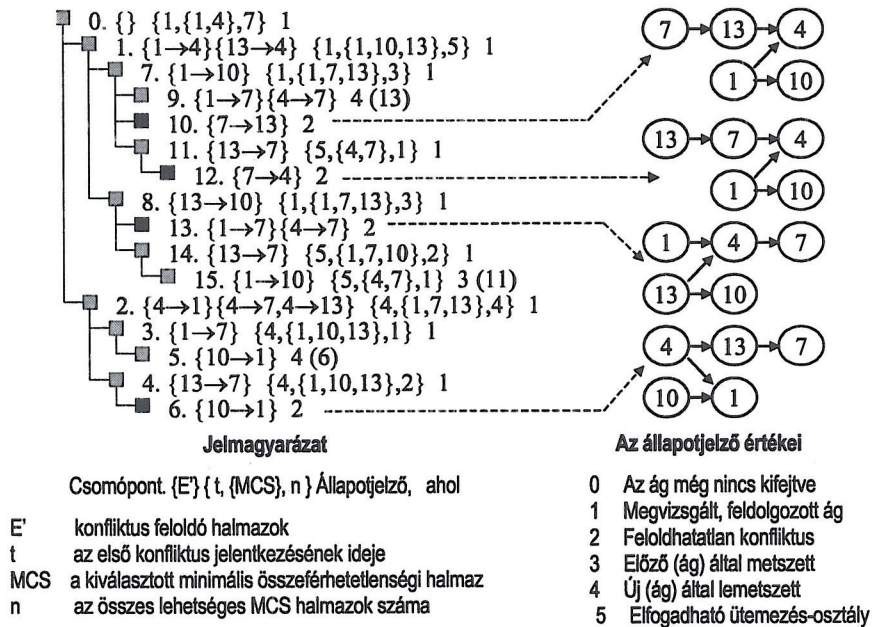
6.1 táblázat: Az *SS* feladat összes lehetséges erőforrás konfliktusának listája

Konfliktus lexicografikus sorszáma	Konfliktus időtartomány	Minimális összeférhetlenségi halmaz	A konfliktusok összes lehetséges javítási lehetőségei
1	[01,09]	{1,4}	{1→4,4→1}
2	[01,09]	{4,7}	{4→7,7→4}
3	[01,09]	{1,7,10}	{1→7,7→1,1→10,10→1,7→10,10→7}
4	[01,07]	{1,7,13}	{1→7,7→1,1→13,13→1,13→7}
5	[01,07]	{4,13}	{4→13,13→4}
6	[01,07]	{1,10,13}	{1→10,10→1,1→13,13→1,10→13,13→10}
7	[01,07]	{7,10,13}	{7→10,10→7,13→7,10→13,13→10}

A 2-es prioritású {4,7} minimális összeférhetlenségi halmaz csak az  $1 \leq t \leq 9$  időpontokban okozhat konfliktust. Ha egy csúcspontot ezen halmaz

szerint kibontunk, akkor két új,  $\{4 \rightarrow 7\}$  és  $\{7 \rightarrow 4\}$  feloldási relációkkal kibővített alcsomópontot kapunk (ebben a sorrendben).

Az algoritmus nulladik fázisának működése (a  $T=14$  határidő esetén) a 6.7 ábra látható. A keresőfa a  $T=14$  határidő metszési szabályaival készült, így levelei vagy metszettek, vagy tovább nem bonthatóak. Ha  $T(i)$ -vel jelöljük az  $i$ . osztály ütemezéseinek legkorábbi határidejét, és a nem bontható (6., 10., 12., és 13.) leveleket megvizsgáljuk, akkor azt tapasztaljuk, hogy  $T(6)=16$ ,  $T(10)=17$ ,  $T(12)=17$  és  $T(13)=16$ . Vagyis egyik sem lehetséges megoldás, következésképp a projekt végrehajtására 14 nap nem elegendő.



6.7 ábra: A 0. fázishoz tartozó fa 2. ágának kibontott változata

Mivel a 6.7 ábra feketével jelölt csúcsaihoz tartozó legkorábbi határidők 15-nél is nagyobbak, ezért ha a határidőt egy nappal meghosszabbítjuk ( $T=15$ ), akkor a felépülő fa mindössze ott térhet el a  $T=14$  esettől, ahol az előbb metszések történtek, azaz a 6., 9. és 15. csomópontokban. Ezek tovább bontásával magát a fa felépítésnek a folyamatát is demonstrálni tudjuk.

Tekintsük először a 15. csomópontához rendelt konfliktus feloldó halmazt:  $E_{15} = \{1 \rightarrow 4, 13 \rightarrow 4, 13 \rightarrow 10, 13 \rightarrow 7, 1 \rightarrow 10\}$ . A közvetlen megelőzési re-



lációknak és az  $E_{15}$  halmaznak megfelelő ütemezések legkorábban a  $t=5$  időpontban okozhatnak erőforrás konfliktust, ahol is legkisebb elemszámú (és egyben az egyetlen) minimális összeférhetetlenségi halmaz a  $\{4,7\}$ . A 6.1 táblázat értelmezéséből már tudjuk, hogy ekkor a 15. csomópont kibontása után a következő két leágazás keletkezik:

$$E_{16} = E_{15} \cup \{4 \rightarrow 7\} = \{1 \rightarrow 4, 13 \rightarrow 4, 13 \rightarrow 10, 13 \rightarrow 7, 1 \rightarrow 10, 4 \rightarrow 7\} \text{ és}$$

$$E_{17} = E_{15} \cup \{7 \rightarrow 4\} = \{1 \rightarrow 4, 13 \rightarrow 4, 13 \rightarrow 10, 13 \rightarrow 7, 1 \rightarrow 10, 7 \rightarrow 4\}.$$

Mindkét keletkezett csomópont konfliktusmentes levél lenne, ha az  $E_{16}$ -hoz tartozó szabályhalmazt az  $E_{13}$ -as, míg az  $E_{17}$ -hez tartozót az  $E_{12}$ -es ág le nem metszené.

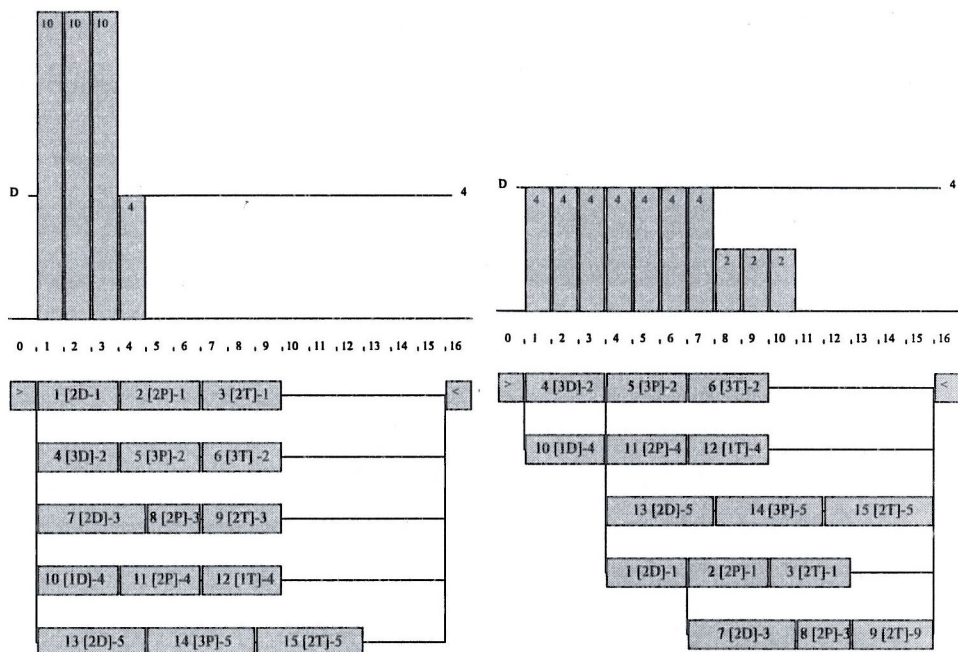
Az  $E_9$  és  $E_5$  halmazokhoz tartozó csomópontok már nem tartalmaznak konfliktusokat. Ezek az ágak most nem is metsződnek le, és a következő osztályokat definiálják:

$$E_9 = \{1 \rightarrow 4, 13 \rightarrow 4, 1 \rightarrow 10, 1 \rightarrow 7, 4 \rightarrow 7\} \text{ és így } T(9)=16, \text{ valamint}$$

$$E_5 = \{4 \rightarrow 1, 4 \rightarrow 7, 4 \rightarrow 13, 1 \rightarrow 7, 10 \rightarrow 1\}, \text{ ahol } T(5)=15.$$

Ennek megfelelően a 9. csomópont nem, míg az 5. csomópont elfogadható ütemezés-osztályt határoz meg, és ezzel egyben az első fázis végére is értünk.

Az  $SS$  feladat tehát olyan szempontból szerencsés, hogy csak egyféle elfogadható ütemezési osztály létezik. Általában azonban több „elfogadható” levél marad a fán, melyek mindegyikére külön-külön kell a második fázist végrehajtani. Az  $SS$  feladat méretéről pontosabb képet kaphatunk, ha tudjuk, hogy a lehetséges ütemezések száma  $|\mathcal{S}|=160\,000$ , míg az 5. ághoz rendelt elfogadható ütemezések halmaza  $|\mathcal{S}^H|=7\,840$  elemből áll, melyet a 6.8 ábrán egyetlen reprezentánssal, az  $E' = \{1 \rightarrow 7, 4 \rightarrow 1, 4 \rightarrow 7, 4 \rightarrow 13, 10 \rightarrow 1\}$  halmazzal jellemezhetünk.



6.8 ábra: Az 1. fázis kiinduló állapota és a végén kapott reprezentáns ES ütemezés

### 6.5.2 Második fázis – Kiegyenlítési feladat

A második fázis lépéseinek kiinduló pontjai a hozzárendelési feladat megoldásaként kapott  $S^H$ -beli ütemezés-osztályok. Az első fázis végeredményeként kapott keresési fa minden olyan leveléhez, amely elfogadható ütemezés-osztályt határoz meg, egy új erőforrás kiegyenlítési feladatot rendelünk hozzá. A kiegyenlítési feladatok az eredeti feladat alábbi módosításával állíthatók elő:

- (1) A feladat határidejéül a 0. fázis utolsó lépésében kapott  $T$  értéket választjuk.
- (2) Az erős erőforrásokat „visszaminősítjük” gyenge erőforrásokká, így az összes erőforrást egységesen gyenge erőforrásként kezeljük.
- (3) Az eredeti feladat közvetlen megelőzési relációinak  $E$  halmazát kibővítjük az adott falevélhez tartozó  $E' = \bigcup \{i \rightarrow j\}$  konfliktus feloldási relációkkal.

- (4) A feladat célvektorául a modellben megadott - állásidőkből és maximális kihasználtsági szintekből álló -  $(IT^{\mathfrak{R}}(S), MU^{\mathfrak{R}}(S))$  vektor minimumát választjuk.

Ha az eredeti modell gyengített változatával állunk szemben (lásd 6.4 fejezet), akkor a módosított feladat célvektorából itt is ki kell hagyni az erős erőforrásokra vonatkozó  $IT^{\mathfrak{R}_H}(S), MU^{\mathfrak{R}_H}(S)$  elemeket, valamint az erős erőforrásokat nem „visszaminősíteni” kell, hanem figyelmen kívül hagyni.

Az eredeti modellnek megfelelően tehát az  $\mathfrak{R}$  erőforrások most mind egyenlően korlátosak, ezért az új kiegyenlítési feladatoknál már nem lesznek erőforrás felhasználási konfliktusok, ami miatt a keresési fák mérete is jelentősen megnő. Az  $r = 1, 2, \dots, k + m$  erőforrásokhoz rendelt  $IT'(S)$  és  $MU'(S)$  célfüggvények Pareto minimumait *hatékony* megoldásoknak nevezzük. A második fázis végrehajtása után kapott hatékony ütemezések egyben az eredeti feladat *egzakst megoldásai* lesznek.

Így minden elfogadható ütemezési osztályhoz egy újabb kiegyenlítési feladatot rendeltünk, amely egy új, de az implicit leszámhlási alapmodellben ismertetett feladattípushoz nagyon közeli probléma megoldását jelenti. Tehetjük mindezt azért, mert a módosított feladatok bármely lehetséges megoldása mindig egyben az eredeti feladat elfogadható megoldás is.

A modell numerikus kezeléséhez többféle implicit leszámhláson alapuló eljárás közül választhatunk. A konkrét megvalósítás során ezek közül mégis a 6.2 fejezetben ismertetett alapalgoritmust alkalmaztuk, annak ellenére, hogy a kapott kiegyenlítési problémák hatékony megoldásainak megkereséséhez az implicit leszámhlási alapalgoritmust módosítanunk kell. A módosításra egyrészt azért van szükség, mert nem egy, hanem kétféle célkritérium (IT és MU) szerinti hatékony megoldásokat keresünk; másrészt az ott remekül bevált „LP relaxálás” vágási szabálya most nem alkalmazható. Helyette azonban más gyorsítási lehetőségeket kereshetünk, de a relaxálás nélküli vágási szabályok is



elég hatékonyak bizonyultak ahhoz, hogy közepes méretű feladatok megoldását is lehetővé tegyék.

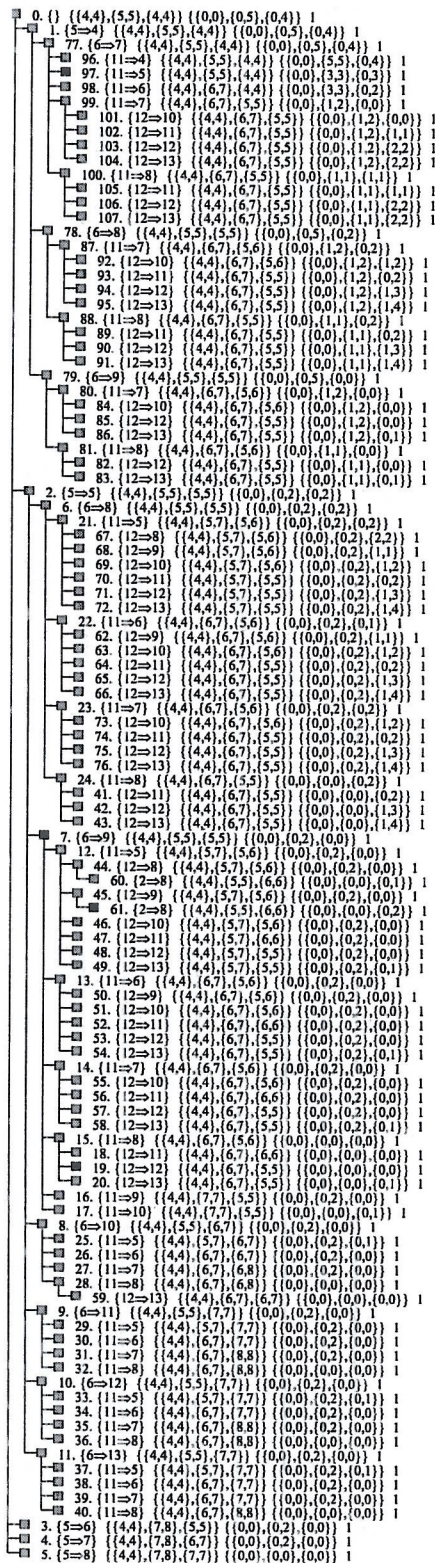
Az implicit leszámplálási alapalgorithmus a lehetséges megoldások  $\mathcal{S}$  halmazának CPM szerinti legkorábbi ES ütemezéséből indult ki. A módosított feladatok esetében ezek az ütemezések azonban már a rendelkezésünkre állnak, hiszen a keresési fa leveleinek reprezentánsai.

## 6.6 A LIT-LMU vágási szabály

A keresőfa csomópontjainak jellemzésére az alapalgorithmus szerinti  $IT^r(ES(F))$  értékeken és a  $LIT^r(F)$  alsó becsléseken túl most az  $MU^r(ES(F))$  értékekre és azok  $LMU^r(F)$  alsó becsléseire is szükségünk lesz. A 6.9 ábrán is látható két újfajta érték közül az első természetes módon számolható, csak a maximális kihasználtsági szintek alsó becslésére kell mód-szert adnunk.

Az  $F = \emptyset$  üres halmaz esetén az adott erőforrásból legtöbbet igénylő tevékenység erőforrás igényét használhatjuk triviális alsó becslésként, azaz  $LMU^r(\emptyset) = \max_{0 \leq i \leq N} R_i^r$ . Ha bizonyos tevékenységek már rögzítve vannak (az  $F$  halmaz által, vagy mert kritikusak), akkor ezek ütemezéséből keletkező  $S'$  csonka ütemezés erőforrás-kihasználtsági szintjeit már használhatjuk a becslések javítására:  $LMU^r(F) = \max(MU^r(S'), LMU^r(\emptyset))$ .

A LIT-LMU vágási szabály a 6.1 lemma analógiájára már könnyen megfogalmazható: Az  $\mathcal{S}^*$  ütemezés-halmaz dominálja az  $\mathcal{S}(F)$  halmazt - röviden  $\mathcal{S}^* \prec \mathcal{S}(F)$  -, ha minden  $r \in \{1, 2, \dots, k + m\}$  erőforrásra nézve teljesül mind az  $IT^r(\mathcal{S}^*) \leq LIT^r(F)$ , mind a  $MU^r(\mathcal{S}^*) \leq LMU^r(F)$  egyenlőtlenség, ahol  $\mathcal{S}^* = ES(\mathcal{S}^*)$  jelöli az  $\mathcal{S}^*$  ütemezés-halmaz reprezentáns elemét. Ilyen esetben ha  $\mathcal{S}^*$  már szerepel csomópontként a keresőfában, akkor az  $\mathcal{S}(F)$ -hez tartozó csomópont kibontása elhagyható.



## Jelmagyarázat

A csomópont sorszámanak feltüntetése után először mindig az  $F$  halmaz bővítését adtuk meg az adott tevékenység időbeli rögzítésével

$$\{i \Rightarrow t\}$$

majd kapcsos zárójelek között először a

$$\{LMU', MU'(ES)\}$$

majd utána az

$$\{LIT', IT'(ES)\}$$

értékek felsorolása következik.

## Állapotjelző

A sor végén a csomópont aktuális állapota látható, melynek jelentése:

- 0 – Generált csomópont
- 1 – Kibontott ág

## Megjegyzések

A soron következő kibontandó csomópont meghatározásakor mindig azt a még ki nem bontott levelet választottuk, ahol a

$$\sum_{r=1}^3 (IT'(ES) + MU'(ES))$$

minimális volt.

Ezen az ábrán - a 6.4 ábrával ellentétben - már nem tüntettük fel azt, hogy melyik csomópontot melyik ág metszette le.

6.9 ábra: A négy hatékony megoldás helye a 2. fázisban felépített, metszések utáni fában

A fa bejárásakor mindig azt az aktív levelet bontjuk ki, amelyre a

$$QC(F) = \sum_{r=1}^{k+m} (IT^r(ES(F)) + MU^r(ES(F))) \text{ kvázi-költség minimális.}$$

Az *SS* probléma megoldását a második fázisnál folytatva egyetlen al-problémát kell megoldanunk. Ennél a módosított feladatnál a közvetlen megelőzési relációk halmaza  $E \cup \{1 \rightarrow 7, 4 \rightarrow 1, 4 \rightarrow 7, 4 \rightarrow 13, 10 \rightarrow 1\}$  lesz. Az alapalgoritmussal most azonban az  $(IT^{\mathfrak{N}}(S), MU^{\mathfrak{N}}(S))$  értékek szerinti Pareto minimumot kell megkeresnünk. Az optimális ütemezésekhez vezető keresőfa 6.9 ábráján feketével jelöltük meg a négyféle hatékony megoldáshoz tartozó levelet, melyeknek adatait a 6.2 táblázatban külön is összefoglaljuk.

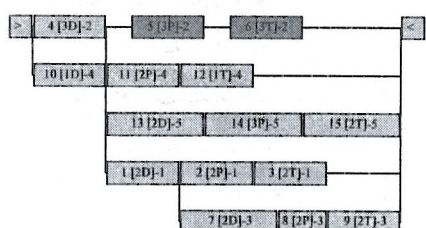
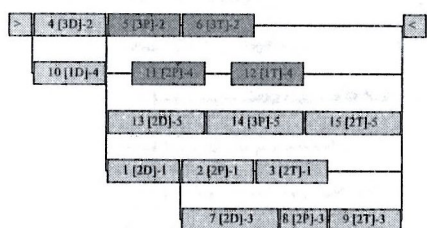
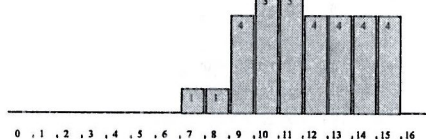
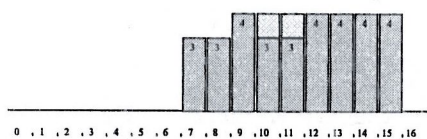
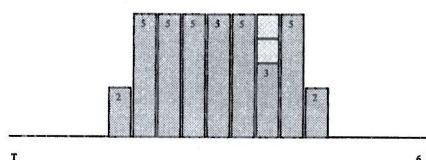
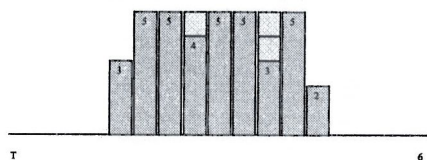
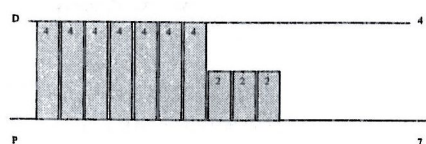
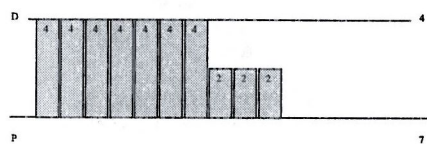
6.2 táblázat: Az *SS* feladat *MU* és *IT* szerint hatékony ütemezései

Sor-szám	$MU_{1..3}$ értékek	$\Sigma MU_i$	$IT_{1..3}$ értékek	$\Sigma IT_i$	$\Sigma$	Rögzített tevékenységek
1	{4, 5, 4}	13	{0, 3, 2}	5	18	{5 $\Rightarrow$ 4, 6 $\Rightarrow$ 7, 11 $\Rightarrow$ 5, 12 $\Rightarrow$ 9}
2	{4, 5, 5}	14	{0, 2, 0}	2	16	{5 $\Rightarrow$ 5, 6 $\Rightarrow$ 9}
3	{4, 7, 5}	16	{0, 0, 0}	0	16	{5 $\Rightarrow$ 5, 6 $\Rightarrow$ 9, 11 $\Rightarrow$ 8, 12 $\Rightarrow$ 12}
4	{4, 5, 6}	15	{0, 0, 0}	0	15	{5 $\Rightarrow$ 5, 6 $\Rightarrow$ 9, 11 $\Rightarrow$ 5, 12 $\Rightarrow$ 9, 2 $\Rightarrow$ 8, 3 $\Rightarrow$ 12}

A második fázisban használt keresőfán a levágott ágakat olvashatósági okokból nem tüntettük fel. A vágatlan fa teljes mérete: **108** (az ábrán megmaradt) + **371** (LIT-LMU metszéssel kivágott) = **479** csomópont.

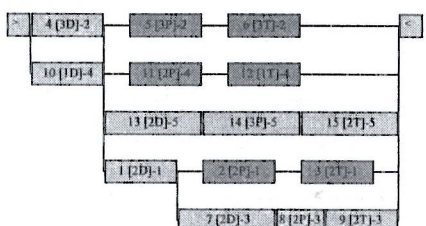
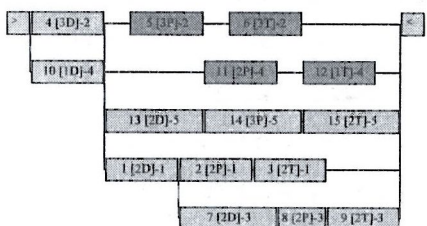
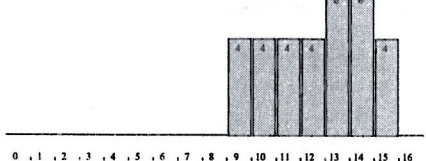
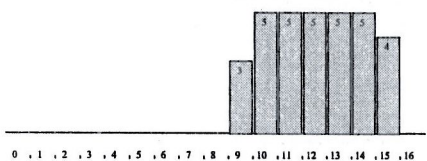
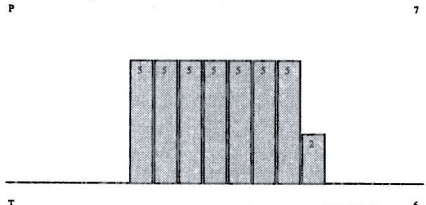
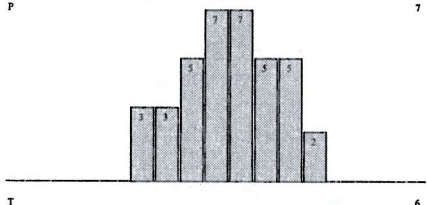
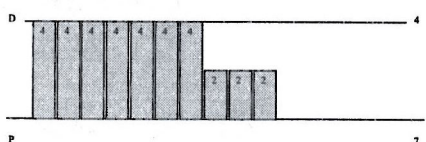
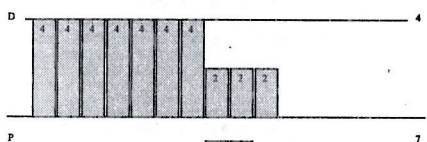
Legvégül az *SS* feladat hatékony megoldásait a 6.10 ábrán adjuk közre.





1. A 97-es ág Pareto optimális ütemezése

2. A 7-es ág Pareto optimális ütemezése



3. A 19-es ág Pareto optimális ütemezése

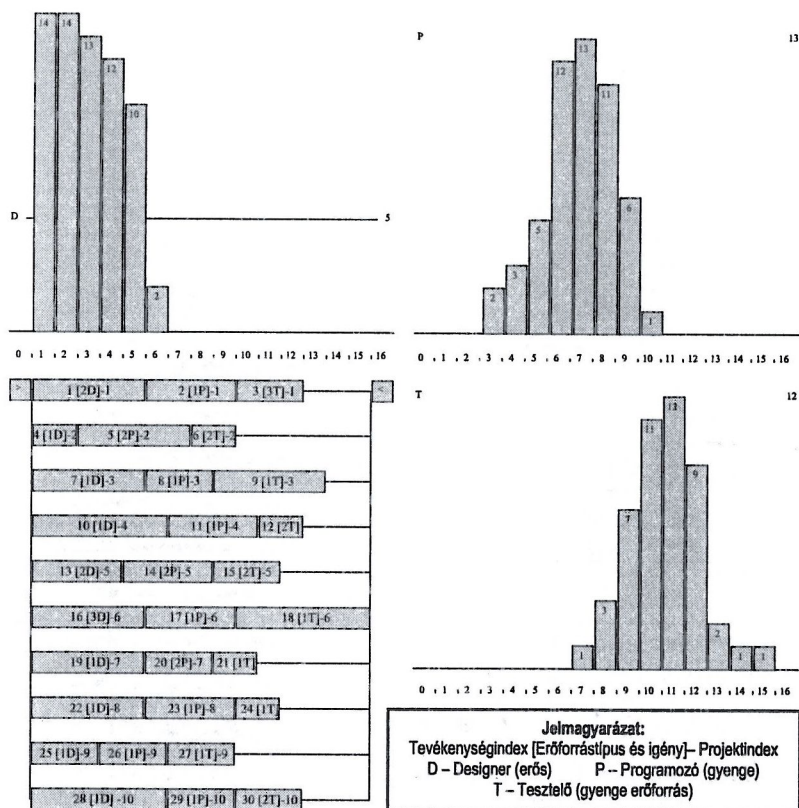
4. A 61-es ág Pareto optimális ütemezése

6.10 ábra: A 2. fázis és egyben az SS feladat Pareto optimális ütemezései

## ERŐFORRÁS KIEGYENLÍTÉSI HEURISZTIKÁK

Tudjuk, hogy az erőforrás kiegyenlítési problémák osztálya NP nehézségű, ezért nagyméretű feladatok egzakt megoldása rendkívül időigényes, ha egyáltalán lehetséges. Amikor egy feladat mérete eléri a kombinatorikus robbanást, akkor megoldás során az alábbi lehetőségeket (vagy ezek kombinációját) vehetjük sorra: (1) az optimális megoldások helyett megelégszünk annak valamilyen közelítésével, vagy (2) az eredeti feladat helyett annak gyengített (relaxált) változatát oldjuk meg, vagy (3) az összes optimális megoldás megkeresése helyett megelégszünk a legelső optimum megkeresésével, vagy (4) újabb vágási szabályok és fabejárési stratégiák felállításával csökkentjük a keresési fa méretét, vagy (5) speciális heurisztikát alkalmazunk, amely kihasználja a feladat speciális szerkezetét, vagy (6) fontosabb elágazásoknál „kézi vezérlésre kapcsolunk”, azaz szakértői döntésre bízunk az algoritmus folytatását.

A 7.1 ábrán ismertetett *LS* (LargeSize.net) feladat valós szoftverfejlesztési környezetből származik, és jól mutatja, hogy mindössze 10 hasonló szerkezetű projekt esetén is mekkora feladattal állunk szemben. Ha közelebbről is megvizsgáljuk a problémát, láthatjuk, hogy a szerkezete igen hasonló az előző fejezetekben bemutatott *SS* problémához. Ez az általánosan használt projekt-szerkezet abból adódik, hogy egy adott szoftvermegrendelést több, kisebb modulra bontva készítünk el. A modulokra bontás lényege az, hogy a feladatot több, kisebb, párhuzamosan elkészíthető egységre bontjuk. A moduloknak az összeépítés előtt ellenőrzésen kell átesniük. A modulok kivitelezése minden esetben a tervezés – kivitelezés – ellenőrzés folyamatrendszerben történik, ahol minden fázishoz csak egyféle erőforrásra van szükség. Jelenleg a vállalatnál 5 fő programtervező (5 fő programozó és 5 fő tesztelő) dolgozik.



7.1 ábra: Az LS probléma tevékenységeinek idő és erőforrásigénye és ES ütemezése

Az *LS* feladathoz tartozó vegyes egészértékű programozási feladat mérete a *Lindo* program szerint a következők:

ROWS = 1164 VARS = 813 INTEGER VARS = 322 (322 = 0/1) NONZEROS = 5286  
 QCP = 0 DENSITY = 0.006 CONSTRAINT NONZ=4882 (4281=+-1) MIN&MAX ABSVALUE = 1 & 16  
 OBJ = MIN NUMBER OF <: 374 NUMBER OF =: 180 NUMBER OF >: 609

A *Lindo* statisztikájából jól látszik, hogy noha a feladat mérete jelentősen (négyzetesen) megnőtt, a mátrix sűrűsége alig változott. Ennek ellenére a feladatnak még a relaxált változata sem bizonyult belátható időn belül megoldhatónak. Ilyenkor esetleg más (például belső pontos módszeren alapuló) szoftver alkalmazása lehet a megoldás.

A feladat speciális szerkezetéből adódóan, a három erőforrásnak megfelelően három konfliktus-centrum keletkezik. Ráadásul ezek párhuzamosan futó tevékenységek, melyek között sehol sincs direkt közvetlen megelőzési reláció defi-



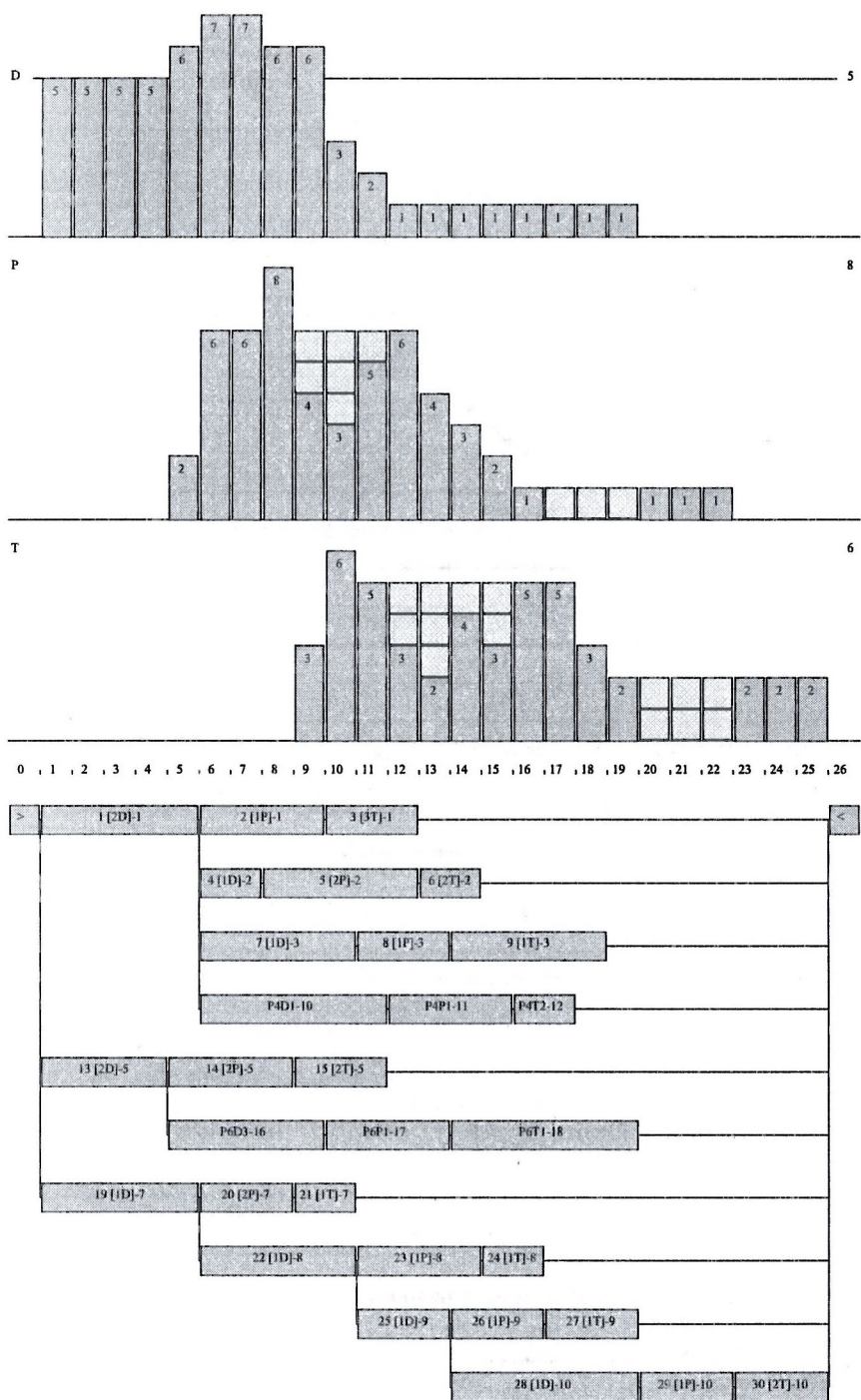
niálva. A tevékenységek mozgathatóságának szabadságfokából származó kombinatorikus ütemezés-szám robbanás adatai a 7.1 táblázatból olvashatóak le. Sajnos a probléma nagysága már az időszak legelején ütemezendő erős erőforrások konfliktusaiban gyökerezik. Ennek következtében a speciális szerkezet adta előnyök kihasználása nehezen látszik megoldhatónak.

7.1 táblázat: A lehetséges ütemezések száma, az erős erőforrás korlát mellőzésével

Projekt	Lehetséges ütemezések száma: $U_p$	Struktúra
P1	20	$\{1 \rightarrow 2 \rightarrow 3\}$
P2	84	$\{4 \rightarrow 5 \rightarrow 6\}$
P3	10	$\{7 \rightarrow 8 \rightarrow 9\}$
P4	20	$\{10 \rightarrow 11 \rightarrow 12\}$
P5	35	$\{13 \rightarrow 14 \rightarrow 15\}$
P6	1	$\{16 \rightarrow 17 \rightarrow 18\}$
P7	56	$\{19 \rightarrow 20 \rightarrow 21\}$
P8	35	$\{22 \rightarrow 23 \rightarrow 24\}$
P9	84	$\{25 \rightarrow 26 \rightarrow 27\}$
P10	20	$\{28 \rightarrow 29 \rightarrow 30\}$
$\prod_{p=1}^{10} U_p$	38 723 328 000 000	

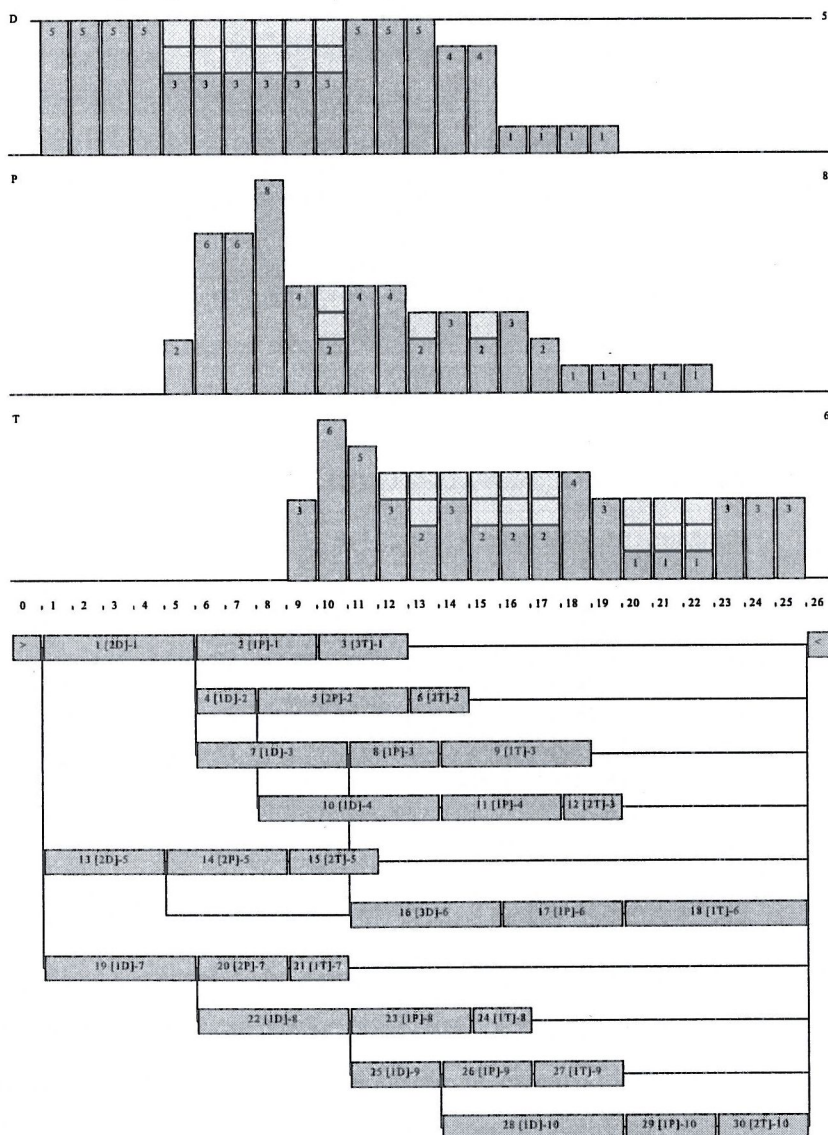
## 7.1 Prioritások alkalmazása

Éppen ezért a nehézségek áthidalásának első lépéseként a „kézi vezérlés” módszerét alkalmaztuk, és megkértük a projektvezetőt arra, hogy a 10 projekt-ről készítsen prioritási táblázatot. A prioritási táblázat adataiból az derült ki, hogy a projektek három független prioritási halmazba sorolhatóak:  $H_1 = \{P_1, P_2, P_3, P_4\}$ ,  $H_2 = \{P_5, P_6\}$  és  $H_3 = \{P_7, P_8, P_9, P_{10}\}$ . Az egyes halmazokon belül a kapott prioritási sorrendet a projektek tervezési fázisai között meghúzott extra közvetlen megelőzési relációkkal írhatjuk le:  $E_2 = \{13 \rightarrow 16\}$ ,  $E_1 = \{1 \rightarrow 4, 1 \rightarrow 7, 1 \rightarrow 10\}$  valamint  $E_3 = \{19 \rightarrow 22, 22 \rightarrow 25, 25 \rightarrow 28\}$ . A prioritási táblázatból származtatott relációikkal kiegészített **LS** feladat CPM szerinti legkorábbi ütemezése a 7.2 ábrán látható. Bár az új **LSP** feladat mérete ennek következtében nagyságrendekkel csökkent, az összes Pareto optimális megoldás megkeresése továbbra is kivitelezhetetlen.



7.2 ábra: Az LS feladat prioritásokkal kiegészített LSP változatának CPM ütemezése

Ha tehát tovább kívánunk lépni, újabb rövidítést kell alkalmaznunk. Mivel az előző fejezetben kidolgozott keresés első fázisa az erős erőforrásokra nézve keres elfogadható ütemezéseket (és a prioritások bevezetésével éppen ezek számát sikerült erősen lecsökkenteni), ezért most érdemesnek tűnik az „első megoldás – jó megoldás” módszerét alkalmazni. Az 1. fázis legelső elfogadható ütemezési osztályához vezető úton kibontott 16 csúcspontú keresési fa paraméterei a 7.4 ábrán láthatóak. A 15. csúcspont osztályához tartozó reprezentáns ütemezést pedig a 7.3 ábrán mutatjuk be.



7.3 ábra: Az LSP feladat D erőforrására vonatkozó első elfogadható ütemezése





7.4 ábra: Az LSP feladat 1. fázisának első elfogadható ütemezését adó keresőfája

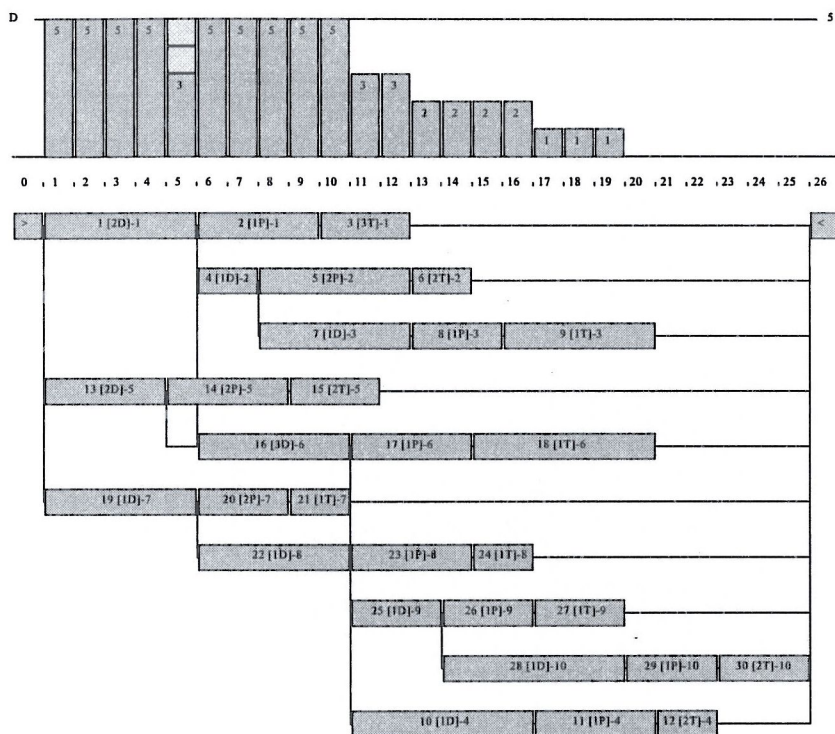
Ezzel az első fázis végére értünk. Ha azonban szükséges, az első fázishoz tartozó keresőfát teljesen ki is bonthatjuk. Jelen esetben a teljesen kibontott fa összesen 396 csúcspontot tartalmaz, és kibontásához 698 CPM ütemezést kell kiértékelni. A teljes kibontás után lehetőségünk van az összes elfogadható ütemezés-osztály megadására is. Az **LSP** feladat esetén 22 féle ilyen osztály létezik, melyeket a 7.2 táblázat tartalmazza, azok fellelési sorrendjében.

7.2 táblázat: Az LSP probléma összes elfogadható ütemezés-osztálya

Sor-szám	Konfliktus feloldási relációk	Sor-szám	Konfliktus feloldási relációk
1	{1→16,4→10,7→16}	12	{1→16,16→4,16→7}
2	{1→16,7→16,10→4}	13	{1→16,16→7,16→10}
3	{1→16,4→10,7→16,13→4}	14	{1→16,4→7,16→10}
4	{1→16,7→16,10→4,13→4}	15	{1→16,4→16,16→10}
5	{1→16,7→16,13→4,16→4}	16	{1→16,16→4,16→10}
6	{1→16,4→10,7→16,13→10}	17	{1→16,4→7,7→10}
7	{1→16,7→16,10→4,13→10}	18	{1→16,4→16,7→10}
8	{1→16,4→16,7→16,13→10}	19	{1→16,7→10,16→4}
9	{1→16,7→16,13→10,16→4}	20	{1→16,7→4,7→10,16→4}
10	{1→16,4→10,16→7}	21	{1→16,4→16,10→7}
11	{1→16,4→16,16→7}	22	{1→16,10→7,16→4}

Az összes elfogadható ütemezés-osztály ismeretében lehetőség nyílik például arra is, hogy megadjuk azt az ütemezést, amely az erős erőforrásra nézve nemcsak elfogadható, hanem az állásidők tekintetében minimális is. A 7.5 ábrán érdemes ellenőrizni, hogy az erős erőforrás állásidejére minimalizált el-

fogadható ütemezés éppen 7.2 táblázat 14. osztályának reprezentáns ütemezésével esik egybe.



7.5 ábra: Az LSP feladat D erőforrására vonatkozó, állásidő szerinti minimuma

Mivel a második fázis keresési fái ennél nagyságrendekkel nagyobbak lesznek (mint azt a 6. fejezetben is láttuk), ezért mindenféleképpen korlátoznunk kell azon 1. fázisbeli falevelek számát, amelyeket a második fázisban majd kibontunk. Kövessük azt a nagy feladatok esetén szokásos megoldási módszert, hogy kizárólag a legelsőnek megtalált elfogadható ütemezési osztályt vizsgáljuk tovább. Ennek megfelelően a 7.2 táblázat 1. osztályát vizsgáljuk részletesebben, és a közvetlen megelőzési relációk halmazát kiegészítjük a  $\{1 \rightarrow 16, 4 \rightarrow 10, 7 \rightarrow 16\}$  halmaz elemeivel.

A 2. fázisban keletkező (IT, MU) keresőfa mérete azonban így is kezelhetetlen. Ezért a növekedését korlátozni kell, így mindössze az első 10 legígéretebb csomópont kibontását engedélyezzük. Az így kapott fa a 7.6 ábrán látható, ahol sorrendben a 0., 2., 24., 28., 35., 36., 61., 70., 83. és 84. ágakat bontottuk csak ki.







Ezzel a korlátozott kibontással valójában fák mélységi és szélességi keresésének egy olyan heurisztikus kombinációját alkalmazzuk, amely várhatóan elég közel kerül az állásidők és a maximális kihasználtság szerinti minimumhoz.

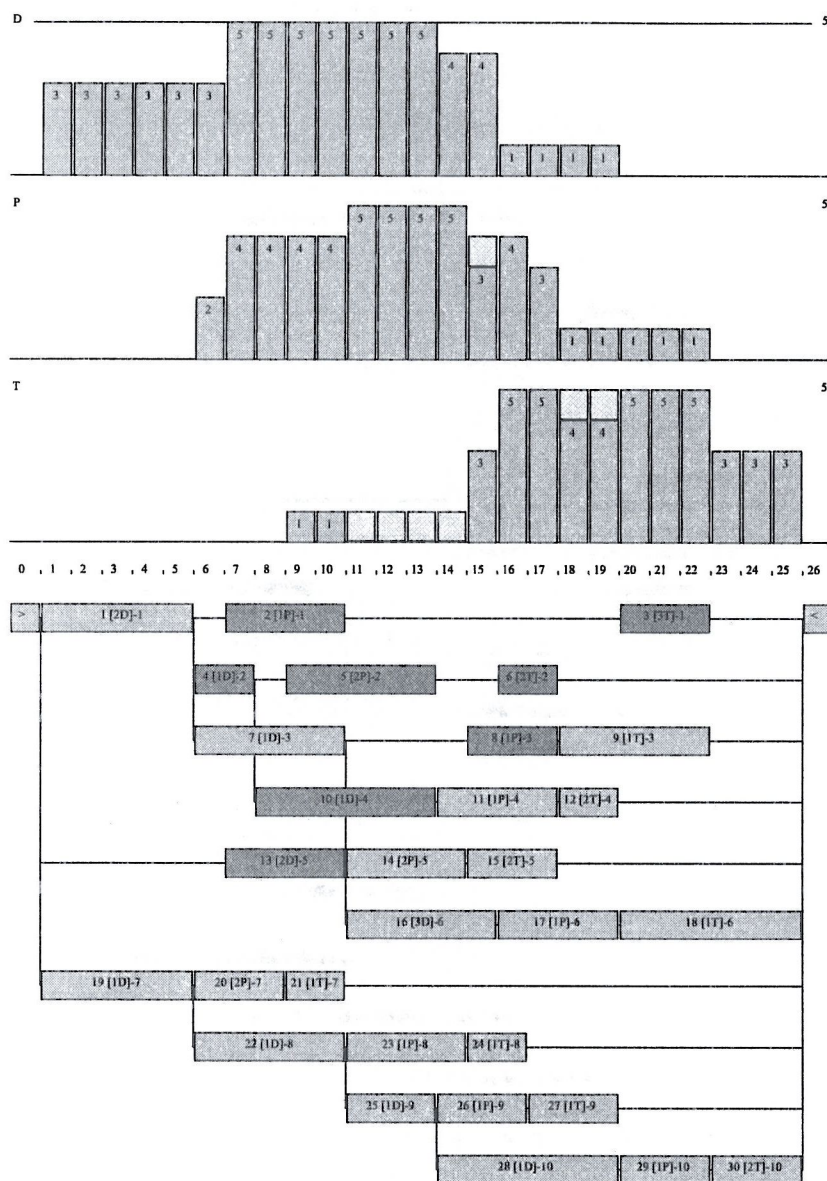
A 7.3 táblázatban külön kiemeltük a korlátozott keresési térben kapott Pareto optimális ütemezéseket. Az itt szereplő közelítő megoldások mindegyikét megvizsgálva, a 88. csomópontához tartozó ütemezést találtuk a legszebbnek, hiszen a célfüggvények összege itt úgy a legkisebb, hogy közben a maximális erőforrás-kihasználtsági szintek a legalacsonyabbak

7.3 táblázat: A második fázis első 10 kibontási lépése utáni hatékony ütemezések

Csomópont	MU <sub>1,2,3</sub> és IT <sub>1,2,3</sub> értékek	$\Sigma$	Tevékenységek kezdeteinek rögzítése
79.	{5,6,4, 8,1,2}	26	{2⇒7,3⇒20,4⇒6,5⇒9,6⇒16,10⇒8,13⇒3}
81.	{5,6,4, 4,1,3}	23	{2⇒7,3⇒20,4⇒6,5⇒9,6⇒16,10⇒8,13⇒5}
83.	{5,6,6, 0,1,4}	22	{2⇒7,3⇒20,4⇒6,5⇒9,6⇒16,8⇒11,9⇒15,10⇒8,13⇒7}
86.	{5,6,6, 0,0,6}	23	{2⇒7,3⇒20,4⇒6,5⇒9,6⇒16,8⇒13,10⇒8,13⇒7}
<b>88.</b>	<b>{5,5,5, 0,1,6}</b>	<b>22</b>	<b>{2⇒7,3⇒20,4⇒6,5⇒9,6⇒16,8⇒15,10⇒8,13⇒7}</b>

Legvégül a 7.7 ábrán a 88. csomópontához tartozó ütemezést adjuk meg, mint az *LS* példán keresztül bemutatott esettanulmányunk többféle heurisztikát is alkalmazó, „közel hatékony” megoldását. A példa azt is jól szemlélteti, hogy a keresőfák keresőfája struktúra sokféle heurisztika alkalmazását teszi lehetővé.

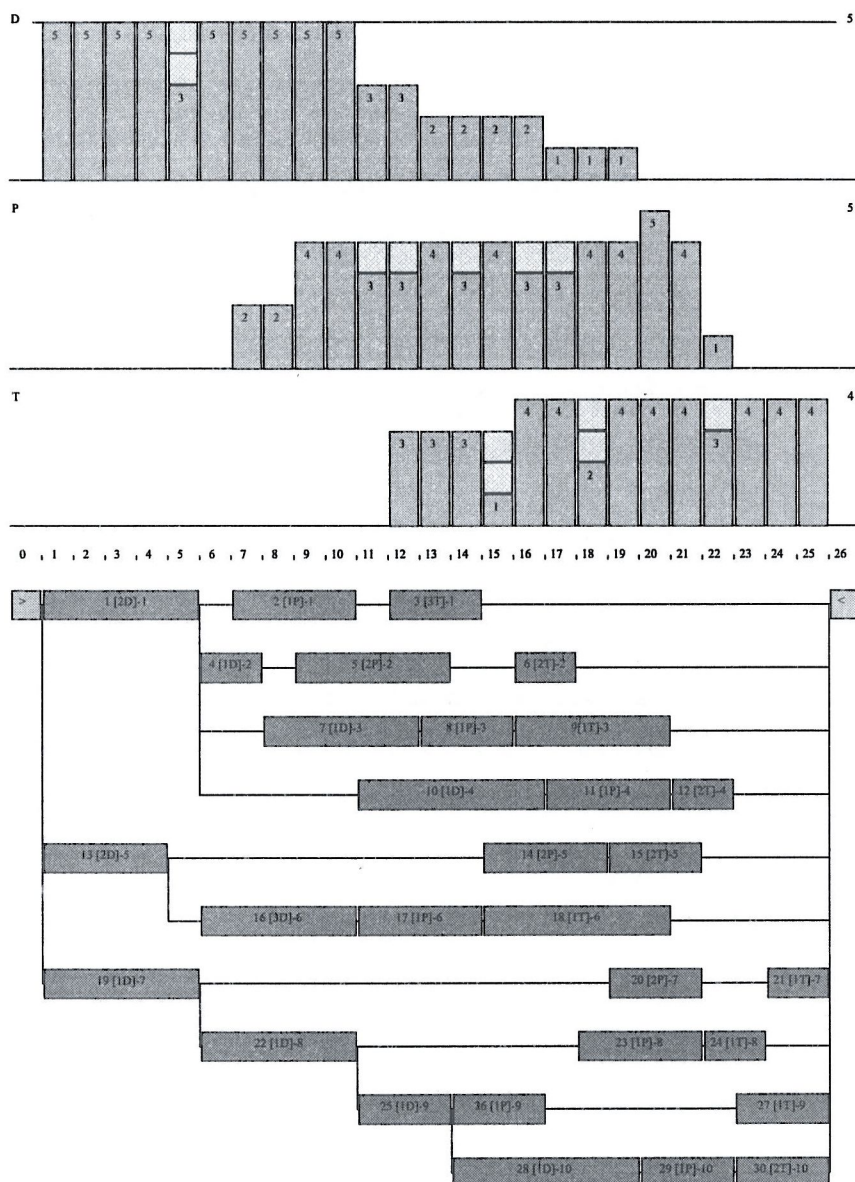
A bemutatott példákban mindenütt legfeljebb egy erős erőforrással találkozhattunk. Ez azonban nem korlátozza az ismertetett modelleknek és megoldási módszereknek, mindössze az ábrák méretkorlátai miatt szorítottunk ilyen esetek vizsgálatára. Hasonlóan nem része a modellnek az a megkötés sem, miszerint minden tevékenység elvégzéséhez csak egyetlen fajta erőforrásra lehet szükségünk. Kisméretű szoftverfejlesztési projektek körében – ahonnan az *LS* probléma is származik – ez az egyszerűsítő feltétel többnyire jól alkalmazható [3.4].



7.7 ábra: Az LSP feladat „legszebb”, „közel hatékony” ütemezése

A fejezet elején szereplő heurisztikus eszközök közül egyedül az új metszési szabályok megalkotásának problémakörére nem tértünk ki eleddig. Most is csak annyit jegyzünk meg, hogy az alapalgorithmusnak nagyobb feladatok megoldására létezik egy „*legfeljebb annyira mozgatható*” (*Less-Shiftable*) vágási szabály kiterjesztése is, amelynek ötlete a tevékenységek rögzítési halmazainak mintájára épül [7.1].

Legvégül az implicit leszámhlási módszer ismeretében természetes módon adódik az időkorlátos megoldáskeresés, hiszen a keresőfa kibontása mindig az aktuálisan legígéretesebb csomópontnál folytatódik. A közelítő megoldást tehát a legkönnyebben a program futási idejének korlátozásával érhetjük el. A 7.8 ábrán láthatjuk azt az ütemezést, amelyet a kétfázisú módszer egy óra alatt a legjobbnak talált. Azt tapasztalhatjuk, hogy egy tesztelő végig, egy programozó pedig egy nap kivételével szabadsággolható a projekt időtartama alatt.



7.8 ábra: Az LSP probléma legjobb ütemezése, a keresést egy óra után megszakítva



## ÖSSZEFOGLALÁS

A dolgozat a projektmenedzsment témakörébe tartozó tevékenység ütemezési problémák területén mutat fel új eredményeket. A középpontba állított (több projektet, többféle erőforrást és célkritériumot is kezelni képes) determinisztikus modell már önmagában is érdekes, gyakorlati jelentőséggel bíró eredmény, de elméleti vonatkozásaiban is fontos elemeket tartalmaz, lévén olyan területeket köt össze (erőforrás hozzárendelés és kiegyenlítés), amiket eddig főleg egymástól függetlenül vizsgáltak. A dolgozat nagy hangsúlyt fektet arra, hogy ne elégedjen meg egy tisztán matematikai modell kialakításával, hanem lehetőség szerint figyelembe vegye a projektmenedzsmenthez kapcsolódó emberi erőforrások kezelésének specialitásait is. Az ütemezések hatékonyságának jellemzésére például éppen ezért olyan mértéket használ (minimális állásidő és maximális kihasználtság), amely a gyakorlati elvárásoknak a legjobban megfelel. Az ismertetett modellek létjogosultságát a dolgozat mind a projektmenedzsment, mind pedig matematikai modellezés oldaláról igazolja, és azok ismertetésén túl sor kerül azok *egzakta megoldási módszereinek* kidolgozására is.

## 8.1 Tézisek

## 1. Tézis

*Ha a rendelkezésre álló erőforrások korlátozottsága alapján eddig használt kategorikus (korlátos vs. korlátlan) besorolás helyett egy korlátozottsági skálát használunk, akkor az erőforrások hatékony felhasználásának modelljei pontosabbá tehetők.*

Igazoltuk, hogy az egyes erőforrások korlátozottsága nem tisztán az adott erőforrás belülről fakadó tulajdonsága, hanem korlátozottságának mértéke erősen

függ a projektben betöltött szerepétől, és így végül magától a projekttől is. A gyengén illetve erősen korlátozott erőforrások fogalmát matematikailag is pontosan definiáltuk. Az így kialakított újszerű erőforrás-osztályok integráns részét képezik a kialakított ütemezési modelleknek. Végül javaslatot tettünk arra, hogy milyen szempontok alapján lehet megítélni egy adott erőforrás rendelkezésre állásának korlátozottsági fokát, egy adott projekt tekintetében.

## 2. Tézis

*A többféle (erősen és gyengén korlátozott) erőforrás esetére, több célkritérium mellett, valamint több projekt környezetben is megadtunk egy általánosan használható determinisztikus ütemezési modellt.*

A dolgozatban ismertetett ütemezési modell azon túl, hogy új elméleti eredmény, a gyakorlatban is közvetlenül alkalmazható. Használhatóságát a szoftverfejlesztési projektek esetére bizonyítottuk, de a módszer más (humán erőforrás intenzív) tevékenységek esetére is könnyen átvihető. Az ütemezések, illetve az erőforrás-kihasználtsági profilok hatékonyságának globális jellemzésére olyan kombinált célfüggvény rendszert alkalmaztunk, amely egyszerre minimalizálja a (gépütemezési feladatoknál ismert fogalom analógiájára épülő) tevékenységi állásidőket és maximalizálja az erőforrás kihasználtságot.

## 3. Tézis

*Kisméretű feladatok esetére megadtunk egy nyolclépéses módszert, amellyel a modell egy egyes egészértékű lineáris programozási feladat (MILP) megoldására vezethető vissza.*

A bemutatott módszer egyszerűsége és automatizálhatósága mellett rendelkezik azzal az előnyös tulajdonsággal is, hogy a visszavezetésként kapott MILP feladat együtthető mátrixa igen ritka. Ennek következtében a kisméretű feladatok egzakt megoldása még viszonylag szerény számítógépes eszközökkel is kivitelezhető.

## 4. Tézis

*Közepes méretű feladatok esetére megadtunk egy kétfázisú implicit leszámítási algoritmust.*

Az ismertetett kétfázisú algoritmus leglényegesebb tulajdonsága az, hogy sikeresen ötvözi az erőforrás hozzárendelési és kiegyenlítési feladatokra alkalmazható megoldási módszereket. Kiemelendő sajátossága továbbá az is, hogy képes az összes Pareto optimális megoldás előállítására, nem csak azok valamelyikének megkeresésére.

## 5. Tézis

*Nagyméretű feladatok esetén megmutattuk, hogy a közepes méretű feladatok megoldására kidolgozott kétfázisú módszer sikeresen ötvözhető a hagyományosan alkalmazott heurisztikus eljárások többségével.*

Bemutattuk, hogy a keresési fa speciális felépítési, vágási és korlátozási szabályainak kialakításával hogyan valósíthatóak meg a különféle heurisztikus megközelítések. Bebizonyítottuk, hogy az algoritmus (speciális halmazainak segítségével) lényegében változatlan formában alkalmas a bázis ütemezés folyamatos karbantartására, finomítására. Példákon keresztül igazoltuk, hogy az általunk ajánlott módszerek valós környezetben is hatékonyan segítik az ütemezési problémák kezelését.

Az illusztrációként felhasznált feladatok a *kis és közepes méretű szoftverfejlesztési projektek* általános jellegzetességeit viselik magukon, de a kidolgozott modellek minden olyan környezetre is könnyen átvihetőek, ahol elegendő a projektfeladatok időben lineáris, tervezés – kivitelezés – ellenőrzés szintű felbontása.

## 8.2 További kutatási irányok

A projektmenedzsment ütemezési funkciója nem ér véget azzal, hogy az akvizíciós fázisban elkészül az adott időpontban ismert adatok alapján készíthető legjobb tevékenységi hálóterv – a bázis terv. Az esetlegesen (szoftverfejlesztés esetén gyakrabban) előforduló határidőcsúszások miatt a *tervet időről-időre újra felül kell vizsgálni*. Szerencsére az ütemezési modellhez itt bemutatott megoldási módszereink mindegyike könnyen módosítható úgy, hogy az ilyen



menetközbeni ütemezés-finomításoknak is eleget tegyen. A kisméretű feladatokra kialakított MILP módszernél ehhez mindössze bizonyos korlátok egyenlőtlenség jeleit kell egyenlőség jelre cserélni. Az implicit leszámítási modell esetében az idő során megváltozott helyzetet könnyen leírhatjuk új, addicionális közvetlen megelőzési halmazok, valamint tevékenységkezdet rögzítési halmazok segítségével. Ezt a lehetőséget a 7. fejezetben a projektvezetői preferencia beépítésekor már illusztráltuk is. Az így kapott feladatok megoldása általában még gyorsabb is lesz, hiszen a feladat mérete az idő előrehaladtával exponenciálisan csökken. Az eredeti modellben elvégzett számítások számos részlete a finomított ütemezésnél is újra hasznosítható, hiszen a keresőfák közös részei változatlanok maradnak.

A több projekt kezelésének lehetősége egyúttal magában foglalja az (akár több projektet is átfogó) mérföldkövek kijelölhetőségének lehetőségét, hiszen a modelljeink a több, független projektet egyetlen nagy „szuperprojektként” képesek kezelni. A mérföldkövek új, üres tevékenységek és néhány megelőzési reláció felvételével (a nyitó- és zárótevékenység analógiájára) könnyedén megadhatóak.

Ha általánosabb fegyverként kezeljük azt, hogy a megoldás menete közben is lehetőségünk van újabb közvetlen megelőzési relációk megadására, akkor feltétlenül érdemes kiemelni a (nagyméretű feladatok esetén alkalmazott) szakértői preferenciák beépíthetőségének jelentőségét. Az erőforrások állásidejének és kihasználtsági szintjének paralel minimalizálása pontosan a folyamatos és egyenletes terhelést jelenti, amely alkalmat ad a blokkokban felszabadult szabad kapacitás hatékonyabb multi-taskolására. De az egybefüggő tétlenségi periódusok egyéb célokra is felhasználhatóak, mint például tanulási folyamatok beépítése, újrafelhasználható komponensek előállítás, refaktoring vagy átfogó minőségellenőrzés.

Legvégül pedig ismert, hogy a szoftverfejlesztés területén kiemelten problémás az egyes tevékenységek időtartamának pontos becslése. Természetesen

módon merül fel az igény, hogy érdemes lenne a modelleket a tevékenységi időkre vonatkozó sztochasztikus illetve fuzzy környezetre is alkalmazni.

## KÖSZÖNETNYILVÁNÍTÁS

A szerző köszönetét fejezi ki a Pécsi Tudományegyetem Közgazdaságtudományi Karán működő Gazdálkodástani Doktori Iskolának, hogy anyagilag támogatta a munka alapját képező kutatásokat és konferencia-részvételeket. Külön megköszönöm témavezetőmnek, Dr. Csébfalvi Györgynek, hogy felkeltette az érdeklődésem az ütemezési problémák irányában, valamint hasznos tanácsaiért, melyek rengeteg tévúttól és idővesztéstől mentettek meg. Köszönettel tartozom továbbá Dr. Farkas Ferencnek, aki temérdek időt áldozott arra, hogy a projektmenedzsmenttel foglalkozó részek kellően erős elméleti megalapozást kapjanak. A dolgozatban szereplő tematikai és gépelési hibák minimalizálásban nagy segítségemre volt Dr. Komlósi Sándor lektori munkája. Végül köszönöm mindazon kollégáim segítségét, akik véleményükkel és ötleteikkel nagyban hozzájárultak a dolgozat értékesebbé tételéhez.



## IRODALOMJEGYZÉK

- [1.1] Duncan WR.: *A Guide to the Project Management Body of Knowledge*. Project Management Institute, Upper Darby (1997)
- [1.2] Cleland DI., William KR.: *Project management Handbook*. Wiley & Sons, New York (1988)
- [1.3] Görög M.: *Általános projektmenedzsment*. Aula Kiadó, Budapest (1999)
- [1.4] Görög M.: *Bevezetés a projekt menedzsmentbe*. Aula Kiadó, Budapest (1993)
- [1.5] Papp O.: *Projekt menedzsment: Projektek tervezése, szervezése, irányítása*. Budapesti Műszaki Egyetem Mérnöktovábbképző Intézete, Budapest (1995)
- [1.6] Nemeslaki A.: *Projekt menedzsment*. Nemzetközi Menedzser Központ, Budapest (1995)
- [1.7] Lock D.: *Project management*. Aldershot Publishing, Gower (1997)
- [1.8] Belout A.: *Effects of human resource management on project effectiveness and success: toward a new conceptual framework*. International Journal of Project Management 16(1998) 21-26
- [1.9] Csébfalvi Gy.: *A New Exact Implicit Enumeration procedure for the Resource Constrained Project Scheduling Problem with Discounted Cash Flows*. MOPTA conference (2002) (megjelenés alatt)
- [1.10] Csébfalvi Gy., Torjai L.: *A New Setup Cost Oriented Bicriteria Resource Leveling Model for Projects*. MOPTA conference poster (2002)
- [1.11] Orlikowski WJ.: *Division among the ranks: the social implications of CASE tools for system developers*. The 10th International Conference on Information Systems, Boston (1990)
- [2.1] Drucker PF.: *Innovation and entrepreneurship*. UK: William Heinemann Ltd, (1985)
- [2.2] Rockart JF.: *The Line Takes the Leadership - IS Management in a Wired Society*. Sloan Management review (1988) 57-64
- [2.3] Thite M.: *Leadership styles in information technology projects*. International Journal of Project Management 18(2000) 235-241
- [2.4] Milosevic DZ.: *Selecting a culturally responsive project management strategy*. Technovation 22 (2002) 493-508
- [2.5] Busby JS, Payne K.: *Issues of organisational behaviour in effort estimation for development projects*. International Journal of Project Management 17 (1999) 293-300
- [2.6] Brooks FP. Jr.: *The Mythical Man-Month, Essays on Software Engineering*. Addison-Wesley, (1975)
- [2.7] Verner JM, Overmyer SP, McCain KW.: *In the 25 years since The Mythical Man-Month what we learned about project management?* Information and Software technology 40 (1999) 1021-1026
- [2.8] Goldratt EM.: *The Theory of Constraints*. North Rivest Press, Inc., (1986)
- [2.9] Goldratt EM.: *What is the thing called theory of constraint and how should it be implemented*. North Rivest Press, Inc., (1990)
- [2.10] Goldratt EM.: *Critical Chain*. North Rivest Press, Inc., (1997)

- [2.11] Chatzoglou PD., Macaulay La.: *A review of existing models for project planning and estimation and the need for a new approach*. International Journal of Project Management 14 (1995) 173-183
- [2.12] Steyn H.: *An investigation into the fundamentals of critical chain project scheduling*. International Journal of Project Management 19 (2000) 363-369
- [2.13] Kruzslicz F., Sramó A.: *Rendszerfejlesztés CASE-eszközökkel*. PTE jegyzet (1998)
- [2.14] Dobák M.: *Szervezeti formák és vezetés*. Közgazdasági és Jogi Könyvkiadó. Budapest (1997)
- [2.15] Metcalfe B.: *Project management system design: A social and organisational analysis*. International Journal of Production Economics 52 (1997) 305-316.
- [2.16] Görög M., Ternyik L.: *Informatikai projektek vezetése* (2001) Kossuth kiadó, Budapest
- [3.1] Kupás T.: *Projektek informatikai támogatása*. Budapesti Műszaki Egyetem, előadás sorozat (2002)
- [3.2] Brooks FP. Jr.: *No silver bullet - essence and accidents of software engineering*. Computer 20 (1987) 10-19
- [3.3] Solingen R., Berghout E., Kusters R., Trienekens J.: *From process improvement to people improvement: enabling learning in software development*. Information and Software technology 42 (2000) 965-971.
- [3.4] Kruzslicz F.: *Szoftverfejlesztési tevékenységek egzakt ütemezése erős és gyenge erőforrás korlátok mellett*. Sigma (2002) megjelenés alatt
- [3.5] Vadapalli A., Mone MA.: *Information technology project outcomes: user participation structures and the impact of organization behaviour and human resource management issues*. Journal of Engineering and Technology Management 17 (2000) 127-151.
- [3.6] W. Humphrey: *Managing Technical People - Innovation, Teamwork, and the Software Process*. Addison-Wesley, (1997)
- [3.7] Heiat A., Heiat N.: *A Model for Estimating Efforts Required for Developing Small-Scale Business Applications*. Journal System and Software 39 (1997) 7-14.
- [4.1] Lockyer K., Gordon J.: *Projektmenedzsment és hálós tervezési technikák*. Kosuth Kiadó, (2000) Budapest
- [4.2] Brucker P., Drexel A., Möhring R., Neumann K., and Pesch E.: *Resource-constrained project scheduling: notation, classification, models, and methods*. European Journal of Operational Research 112 (1999) 3-41
- [4.3] Elmaghraby SE.: *Activity nets: A guided tour through some recent developments*. European Journal of Operational Research 82 (1995) 383-408
- [4.4] Farkas M.: *Matematikai kislexikon*. Műszaki Könyvkiadó, Budapest (1979) 311-312
- [5.1] Csébfalvi Gy.: *Optimális erőforrás kiegyenlítő modellek tevékenységi hálókra*. Habilitációs értekezés, PTE Pécs (2001)
- [5.2] Csébfalvi Gy., Konstantinidis P.: *A new exact resource balancing procedure for the multiple resource-constrained project scheduling problem*. Proc. APMOD Extended Abstracts, Limasol, (1998) 11-13

- [5.3] Csébfalvi Gy., Konstantinidis P.: *A new resource leveling procedure for the multiple resource-constrained project scheduling problem*. Decision Sciences Institute 5th International Conference, Athens, Greece, (1999) 1723-1725
- [5.4] Csébfalvi Gy.: *ProMan v2.0 User's Manual*. University of Pécs (1998-2002)
- [5.5] Schrijver, A.: *Theory of linear and integer programming*. John Wiley, Newyork (1987) 266-267
- [5.6] Chaudhuri S., Walker R.A., Mitchell J.E.: *Analyzing and exploiting the structure of the constraints in the ILP approach to the scheduling problem*. IEEE Trans. Very Large Scale Integration Syst. 2 (VLSI 4 1994) 456-467
- [5.7] Al-Khayyal F.A., Falk J.E.: *Jointly constrained biconvex programming*. Mathematics of Operations Research 8 (1983) 273-286
- [6.1] Csébfalvi Gy.: *A fast exact solution procedure for the multiple resource-constrained project scheduling problem*. Proc. APMOD '98 Extended Abstracts, Limasol, (1998) 11-13
- [6.2] Csébfalvi Gy.: *A new multi-criteria resource leveling procedure for the multiple resource-constrained project scheduling problem*. Proc. EURO XVII, (2000)
- [6.3] Csébfalvi Gy., Konstantinidis P.: *Egy implicit leszámításon alapuló új erőforrás kiegyenlítő eljárás*. Szigma XXIX. (1998) 43-52
- [6.4] Csébfalvi Gy.: *Egy optimális erőforrás kiegyenlítő eljárás tevékenységi hálókra*. „Új utakon a magyar operációkutatás” tanulmánykötet Dialóg Campus, (1999)
- [6.5] Cormen T.H., Leiserson C.E., Rivest R.L.: *Algoritmusok*. Műszaki Könyvkiadó Budapest (2001) 208-217
- [6.6] Hromkovič J.: *Algorithmics for Hard Problems*. Springer-Verlag Berlin (2001) 173-208
- [7.1] Kruzsliz F.: *A New Exact Resource Allocation Model with Hard and Soft Resource Constraints*. OR2002 conference selected papers Klagenfurt (2002) 223-228







